

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

THIS PAGE BLANK (USPTO)

JP99/03451

PCT/JP99/03451

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

REQD 02.08.9999	
WIPO	PCT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

E.K.U.

出願年月日
Date of Application:

1998年 6月26日

出願番号
Application Number:

平成10年特許願第195108号

出願人
Applicant(s):

有限会社バリアフリー

09/720069

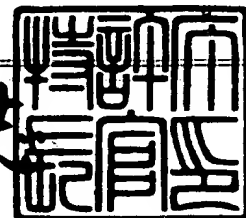
PRIORITY
DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

1999年 8月19日

特許庁長官
Commissioner,
Patent Office

伴佐山 建志



出証番号 出証特平11-3057958

【書類名】 特許願

【整理番号】 P100626

【提出日】 平成10年 6月26日

【あて先】 特許庁長官 殿

【発明の名称】 機能拡張装置および機能拡張方法ならびに機能拡張プログラムを記録した記録媒体

【請求項の数】 15

【発明者】

【住所又は居所】 神奈川県横浜市磯子区中原二丁目13番19号フラット磯子201号

【氏名】 林 毅

【特許出願人】

【郵便番号】 999-99

【住所又は居所】 神奈川県横浜市磯子区中原二丁目13番19号フラット磯子201号

【氏名又は名称】 有限会社バリアフリー

【代表者】 林 毅

【連絡先】 045 (776) 3524

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【書類名】 明細書

【発明の名称】 機能拡張装置および機能拡張方法ならびに機能拡張プログラムを記録した記録媒体

【特許請求の範囲】

【請求項 1】 既存WWWサーバがURLを処理するに際して意図的にエラーを誘発するエラー誘発手段と、前記既存WWWサーバが記憶手段に保持するデータのうち少なくともURLパス部を利用して処理を行う文字列処理手段と、前記エラー誘発手段によりエラーが発生した時に前記既存WWWサーバから前記文字列処理手段へと処理を分岐させる分岐手段とを備える機能拡張装置。

【請求項 2】 既存WWWサーバがURLを処理するに際して意図的にエラーを誘発するエラー誘発手段と、前記既存WWWサーバが記憶手段に保持するデータのうち少なくともURLパス部を利用して処理を行う文字列処理手段と、前記エラー誘発手段によりエラーが発生した時に前記既存WWWサーバから前記文字列処理手段へと処理を分岐させる分岐手段とを備える機能拡張装置であって、前記エラー誘発手段により発生するエラーの程度またはタイミング等を変更するための条件変更手段を備える機能拡張装置。

【請求項 3】 前記エラー誘発手段、前記文字列処理手段、ならびに前記分岐手段のうちの 1 または 2 以上の手段を、前記既存WWWサーバの固定的部位を何ら変更することなく備える請求項 1 または請求項 2 記載の機能拡張装置。

【請求項 4】 前記URLパス部を含むURLは、WWWクライアントのURL入力表示欄で入力されたクライアント側入力文字列に由来し、前記文字列処理手段はWWWサーバに対してどのような形式をもっても伝達されない箇所以外は前記クライアント側入力文字列が入力された形式の表記にて獲得する獲得手段を備える請求項 1 または請求項 2 記載の機能拡張装置。

【請求項 5】 前記URLパス部を含むURLは、WWWクライアントのURL入力表示欄で入力する以外の方法で入力されたまたは設定されたクライアント側入力文字列に由来し、前記文字列処理手段はWWWサーバに対してどのような形式をもっても伝達されない箇所以外は前記クライアント側入力文字列が入力された形式の表記にて獲得する獲得手段を備える請求項 1 または請求項 2 記載の機能拡張装置。

【請求項 6】 前記 URL パス部は漢字、ひらがなまたはカタカナを含み、CGI プログラム等へのパラメータ伝達を指示するものであると RFC により定義された文字列をその定義の通りに用いる用途では含まない請求項 4 または請求項 5 記載の機能拡張装置。

【請求項 7】 前記 URL パス部は

- (1) 日本語や他の国または地域の言語の語句を含むか、
- (2) 特定の学術分野等で用いられている記号類を用いた表記を含むか、または、
- (3) コンピュータ言語の命令やデータを含むか

のいずれかあるいはそれらの 2 以上の組合せである請求項 4 または請求項 5 記載の機能拡張装置。

【請求項 8】 前記 URL パス部は検索を指示する文字列が含まれている請求項 4 または請求項 5 記載の機能拡張装置。

【請求項 9】 前記文字列処理手段は、

- (1) RFC により特別の意味が設定されている文字列のうちの少なくとも 1 についてこれを本来の文字そのままのデータとして処理する特別文字列普通解釈手段か、または、
- (2) 予め定義されたまたは随時定義する文字列を RFC により特別の意味が設定されている文字列と同等に処理する普通文字列特別解釈手段

のうちの少なくとも一方を備える請求項 4 または請求項 5 記載の機能拡張装置。

【請求項 10】 既存情報処理手段が情報処理するに際して意図的にエラーを誘発するエラー誘発手段処理ステップと、前記既存情報処理手段における情報処理結果の一部または全部を利用して前記既存情報処理手段の外で処理を行う外部情報処理手段と、前記エラー誘発手段によりエラーが発生した時に前記既存情報処理手段から外部情報処理手段へと処理を分岐させる分岐手段とを備える機能拡張装置。

【請求項 11】 前記エラー誘発手段により発生するエラーの程度またはタイミング等を変更するための条件変更手段を備えている請求項 10 記載の機能拡張装置。

【請求項 12】 既存 WWW サーバが URL を処理するに際して意図的にエラーを

誘発するエラー誘発設定ステップと、前記既存WWWサーバが記憶部に保持するデータのうち少なくともURLパス部を利用して処理を行う文字列処理ステップと、前記エラー誘発設定ステップによりエラーが発生した時に前記既存WWWサーバから前記文字列処理ステップへと処理を分岐させる分岐処理ステップとを備える機能拡張方法。

【請求項13】 既存WWWサーバがURLを処理するに際して意図的にエラーを誘発するエラー誘発設定ステップと、前記既存WWWサーバが記憶部に保持するデータのうち少なくともURLパス部を利用して処理を行う文字列処理ステップと、前記エラー誘発設定ステップによりエラーが発生した時に前記既存WWWサーバから前記文字列処理ステップへと処理を分岐させる分岐処理ステップとを備える機能拡張プログラムを記録した記録媒体。

【請求項14】 既存情報処理手段が情報処理するに際して前記既存情報処理手段における情報処理結果の一部または全部を利用して前記既存情報処理手段の外で処理を行う外部情報処理手段と、エラーが発生した時に前記既存情報処理手段から外部情報処理手段へと処理を分岐させる分岐手段とを備える機能拡張装置。

【請求項15】 既存WWWサーバがURLを処理するに際して前記既存WWWサーバが記憶手段に保持するデータの一部または全部を利用して処理を行う文字列処理手段と、エラーが発生した時に前記既存WWWサーバから前記文字列処理手段へと処理を分岐させる分岐手段とを備える機能拡張装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、既存の情報処理手段の機能を拡張する装置、方法ならびにこの方法を備えたプログラムに関し、特にWWWサーバにおけるURLの解釈を柔軟に行うことによりURLの表現力や利用価値を高めることに関する。

【0002】

【従来の技術】

インターネットにおけるWWW (world-wide web)サービス (「Web (ウェブ) サー

ビス」とも言う。)の利用形態の例を図2に示す。WWWサーバ(210)とWWWクライアント(202)とは通常、(204)または(208)に示すような小規模なまたは広域のネットワークを経由して接続される。両者の間には、両者間の見かけ上の通信速度の改善等を目的としたWWW中継処理手段(206)が介在する場合がある。WWWサーバ(210)の一般的な構成と関連データを図6に示す。

【0003】

WWWサービスを実施する際にWWWサーバ(210)およびWWWクライアント(202)の両者は、インターネットにおける通信サービスの標準等を定めたRFC(request for comments)なる文書集においてHTTP(hyper text transfer protocol)として定められた通信規約に従ってサービスを実施する。

【0004】

図3Aは、WWWサーバ(210)とWWWクライアント(202)との間でやり取りされる通信処理を簡単に示したものである。基本的には、

(1)WWWクライアント(202)がWWWサーバ(210)に対してHTTPリクエスト(HTTPに基づく資源アクセス要求)を出し、

(2)WWWサーバ(210)がその応答(HTTPレスポンス)として所定の資源をWWWクライアント(202)に返す；

というものである。WWWクライアントがアクセスを所望するWWWサーバ上の資源等を表す方法として図4に示すURL(uniform resource locators)が用いられる。URLもまたRFCの1つとして定められた規約である。

【0005】

図3Bは、WWWクライアント(202)がWWWサーバ(210)に対してHTML(hyper text markup language)ファイルをアクセスしている様子を示したものである。上述したように

(1)WWWクライアント(202)はWWWサーバ(210)に対してHTTPに従いファイル名「index.html」をURLパス部(図4A参照)に含んで送信要求を行い、

(2)WWWサーバ(210)がその応答としてindex.htmlをWWWクライアント(202)に送信している。

なおHTTPでは、ファイルの送信要求以外にもプログラムの実行要求や動作確認等が指定可能である。

【0006】

上記の手順に従ってファイルをアクセスする様子を図5に示すブラウザ（WWWクライアントの機能を実現するプログラムの呼称の1つ。「Webブラウザ」とも言う。）の画面インターフェース例を使って説明すると、次のようになる。

【0007】

(1)ブラウザの画面インターフェースのURL入力表示欄（502）に図4Aに示すような文字列（これが「URL」である。なお、これは一例に過ぎない。）をブラウザのキーボードインターフェース等により入力してアクセス実行キー（図示せず）を押すことによりWWWクライアント（202）は、URLホスト部（図4A参照）で指定したWWWサーバ（210）へURL等（必要に応じてブラウザの種別等の補助情報を付加する場合がある。）をHTTPリクエストとして送信する。この例ではURLパス部（図4A参照）はアクセス対象とするファイル名を含む。

【0008】

(2)前記HTTPリクエストを受けてWWWサーバ（210）は、URLその他の情報を適宜処理する。URLパス部は直接的または間接的にアクセス対象となるファイルの所在を示している（図19参照）。

WWWサーバ（210）はURLパス部の内容に従って

WWWサーバ（210）内のファイルアクセス部（616）を用いて

WWWサーバ（210）内のファイル保持部（620）から所定のファイルを読み出す。

ファイル保持部（620）の内部構造は図8のようになっており、ファイルが記憶されている。なお、この図においては

access.cgi、

index.html、

sales.html、

takaoka.txt、

tanaka.txt、

tech.html、

の6のファイルが例として示されている。必要に応じて他のファイルが配置されたり消去される等の処理が行われる。

【0009】

(3)読み出しに成功したならばWWWサーバ(210)は、その結果(ファイル内容)をHTTPレスポンスとしてHTTPリクエストを発したアクセス元のWWWクライアント(202)へ送信する。HTTPレスポンスにはアクセスの成否等を示すステータスコード等を含む。なお、読み出しに成功しなかった場合は、WWWサーバ(210)は、読み出しのエラーを示すステータスコード等をHTTPレスポンスとする。

【0010】

(4)WWWサーバ(210)における読み出しが成功した場合にはWWWクライアント(202)は、その内容をファイルの形式に応じて受信データ表示欄(504)に表示する。

【0011】

図9に示すのは「〇×商会」(仮名。〇×商会に対応するドメイン名は「ox.com」とする。)のホームページの例である。

【0012】

トップページの画面(902)は、(902)に示すようにブラウザのURL入力表示欄(502)に対して

http://www.ox.com/index.html

と入力してアクセスを実施した結果、WWWサーバ(210)のファイル保持部(620)内に存在するファイル「index.html」(図8参照)がHTMLの表示規約に則ってブラウザの受信データ表示欄(504)に表示されている様子である。

【0013】

トップページに対応しているファイル

index.html

の中には〇×商会の営業部と技術部の両ページへのリンク情報が記述されている

。営業部のページへのリンク部分(904)をマウスでクリックする等した場合は

営業部のページ画面（908）が、
技術部のページへのリンク部分（906）をマウスでクリックする等した場合は
技術部のページ画面（910）がそれぞれ表示される。

上記2のページのURLはそれぞれ

<http://www.ox.com/sales.html>、

<http://www.ox.com/tech.html>

である。

【0014】

以上が従来技術によるWWWサービスの実施の形態の一例である。○×商会は、一
応は、自己が望む内容のホームページを設営できているのかもしれない。

【0015】

【発明が解決しようとする課題】

しかしながら、HTTPやURLの規約に準拠したこれまでのURL（このようなURLを以
降、「TrURL」（Traditional URL）と言う。）ではURLパス部に漢字等の全角文
字を含むことができなかった。

そのため、本当は○×商会が図9のURL入力表示欄（901）に示す表現方法に
代えて図10のURL入力表示欄（1001）、（1007）、（1009）に示
すような漢字、ひらがら、カタカナを含むURL（このようなURLを以降、「KURL」
（Kanji URL）と言う。）を用いたくてもできなかった（またはできないと考え
られていた）。

【0016】

少なくとも資料1（出典は後述）の雑誌の記事は、

「ブラウザに表示させるURLに日本語を使うことはできません。」

としている。当該雑誌の次号以降で訂正等がなされていないことを考え併せれば
、これがインターネット関連業界の標準的認識と考えてほぼ良いであろう。すべ
てのブラウザがHTTPの規約どおり実装されていればこの記述は妥当である。ところ
が実際の状況はやや異なる。

【0017】

発明者が実験した範囲では、HTTPには反するかもしれないものの、開発者または

バージョンが異なる2以上のブラウザにおいて、URL入力表示欄(502)に漢字等を入力することは可能である。ただし、それ(漢字等を含んだKURL)をWWWサーバに伝達する際の処理方法は異なっている。

【0018】

このため、通信するWWWクライアント(202)とWWWサーバ(210)の双方が同一の漢字コード体系を用いているケースや下記のような対処を行ったケースでは、KURLがあたかも正常に使えているように処理することが可能である。

【0019】

すなわち、(他の方法もあるかもしれないがここで説明するのは、)利用される漢字コード体系の別をP、HTTPに基づくエンコード処理の有無やエンコード処理後の文字列の相違の別をQとする時、 $P \times Q = R$ 通りの表記によるファイルをディレクトリ領域(図8A参照)に設定する；という対処方法である。仮に $P=3$ 、 $Q=2$ とした場合、1のURLに対して6のファイルを用意することになる。

【0020】

ただし前記の方法には欠点がある。WWWサーバが動作しているコンピュータにおけるOS(オペレーティングシステム)の種類によっては、特定の漢字コード体系における特定の文字を含む場合は、対応するファイルを設定できない可能性がある。それは——漢字等の全角文字は多くの場合2バイト(16ビット)が1文字に対応する。前記の方法では、外見上は漢字等を用いたファイル名が設定できているように見えながら、実は、2の1バイト文字(ここで「1バイト文字」は7ビットまたは8ビットの空間を持つ文字コード集合。例えばASCIIコード。)をただ連ねただけで(それ以外の配慮をせずに)1の全角文字を疑似的に実現している場合がある。このような漢字処理方法においては、当該OSがファイル名には用いないとしている1バイト文字(例えばあるOSにおいては「/」(スラッシュ))を含むような全角文字は正しく取り扱うことができないおそれがある。

【0021】

仮に上記のような現象が発生する可能性が低いとしてこれを許容するとしても、1のURLに対して6のファイルを用意するのにはファイルの設置や管理に手間が掛かる。それに、もしもKURLの処理方式の異なるブラウザの数が増えて $Q=5$ とな

った場合には、P=3のままとして、1のKURLに対して15ものファイルを用意しなければならない。PやQは将来の増加の可能性があるので、このような場当たり的な対処方法は運用面で大変手間の掛かる方法であると言える。しかも、上述したような不完全性を備えている。

【0022】

TrURLには漢字等の全角文字以外にも表記上の制限がある。

まず、TrURLで使用可能な文字コード集合は最大でも7ビットの文字コード集合であるASCIIコードだけである。

上述した全角文字を含め、2バイト以上の領域を用いる日本語、韓国語、中国語等用の多バイトのコード体系は使えない。ヨーロッパを中心に用いられている「ISO 8859-1」なるコード体系をはじめとする8ビット（1バイト）の文字コード集合も使えない。

【0023】

TrURLには更に制限がある。

ASCII文字コード集合のうち利用者側の意志で自由に使えない文字（コード）が複数存在する。具体的には、

- (1)「非図形文字」である文字群（1102）、
- (2)「Unsafe」である文字群（1104）
- (3)「Reserved」である文字群（1106）

がある。正確な意味はURLに関するRFCを参照されたい。以下では例を用いて説明する。

【0024】

(1)の「非図形文字」とは紙に印刷できない特殊な文字（キャラクタ）のことである。この用法で言うところの「図形」とは「a」や「=」などの表示可能な文字のことであり、算数や数学で慣用するところの円や三角形等の「図形」とは意味合いが異なるので注意されたい。

非図形文字の一例としては、ベル音（C言語の16進数表現で「0x07」というコードが割り当てられているもの。）を発生される等の制御キャラクタ（機能キャラクタとも呼ばれる。）がある。これらは特定の機能を示す「文字」であるが、そ

れらは「a」という文字とは違い印刷して表記することはできないので、「URL」を成す文字列としてそのまま「表記する」ということはできない。対策としてRFCでは、所定の方法によりエンコードして表記する手段を提供している。前記「ベル音」であれば「%07」と表記することになっている。

【0025】

(2)の「『Unsafe』である」とは、利用形態によって期待したように使える場合もあるし期待したように使えない場合もある；という意味である。利用するWWWサーバ（またはブラウザ）に設定された動作条件によって振る舞いが増える可能性があるもの；と言い替えても良い。よって、用いないのが望ましいか、用いる際には注意が必要となる。

一例として「」を挙げることができる。「tom」なる表現は、あるWWWサーバにおいては利用者「tom」が所有するディレクトリを意味し、別のWWWサーバにおいては単に長さ4バイトの文字列を意味する場合がある。

【0026】

(3)の「『Reserved』である」とは「予約済みの」という意味である。特定の意味に用いるものとして規約上予約されたものである。

具体的には、URLホスト部とURLパス部とを分離したり（ディレクトリ等の）階層構造を表現するための「/」を例として挙げることができる。

URLパス部における「dir1/dir2/file」という表記は、「dir1」なる構造（ディレクトリ）の下位にある「dir2」なるディレクトリの下位にある「file」なるファイルを示している。同様に「3/4」という表記は、「3」なるディレクトリの下位にある「4」なるファイルを示している。

【0027】

TrURLにおけるこれらの表記上の制限のため、TrURLを意志表示手段と考えた場合、意志表示手段としてのその表現能力は低いと言わざるを得ない。例えば英語の文章を普段用いているように表記することができないからである。日本語等も同様である。

【0028】

例えば図12Aに示す英文の疑問文風のURLは、RFCの規約上は図12Bに示すUR

Lと意味や処理結果は変わらない。つまり、URLとしては図12Aは図12Bと等価であり、図12Aにおける「?」は英文の疑問符として価値すなわち文字または符号としての価値を持っていない。

【0029】

また、図12Cのようにごくふつうの英文表記「How much?」をそのままURLパス部に含ませることもできないもしくは望ましくない。なぜなら、「How」と「much」との間にある空白文字（16進数表現で0x20）と末尾の「?」はいずれも「Unsafe」な文字であるからである。これは大変不自由なことである。とりわけURLの表示を広告手段と考えた場合、表現能力の自由度の低さは障害となる。

【0030】

「非図形文字」である文字群（1102）、
「Unsafe」である文字群（1104）または
「Reserved」である文字群（1106）や、
ヨーロッパ特有の文字や漢字等の全角文字を含む8ビットまたは16ビット（またはそれ以上）の文字コード集合の利用について少しでも制限が解除されたURL（このようなURLを以降、「FxURL」（Flexible URL）と言う。FxURLは前記KURLを含む概念である。）があれば表現能力の自由度は向上するので好ましい。

【0031】

そこでこの発明は、WWWクライアントからWWWサーバに伝達されたURLの中のURLパス部を柔軟に解釈して多様な文字列処理をすることを可能にすべく、既存のWWWサーバに取り付け可能な機能拡張装置および機能拡張方法ならびに前記機能を備えた機能拡張プログラムを記録した記録媒体を提供することを目的とする。

【0032】

【課題を解決するための手段】

請求項1の機能拡張装置は、意図的にエラーの発生を誘うエラー誘発手段を用いてURLに関する処理を既存のWWWサーバから既存のWWWサーバ以外の処理手段である文字列処理手段（この文字列処理手段はこの発明により新規に用意する手段である。）へと移している。その際、既存のWWWサーバが情報処理した結果として記憶された記憶手段の情報を利用している。

【0033】

これにより、既存のWWWサーバでは実現が困難であったURLパス部の柔軟な解釈が可能となる。この発明の一実施形態による機能拡張手段等を備えたWWWサーバの処理の流れを図1に示す。

【0034】

前記文字列処理手段における情報処理内容を限定する事由は無く、URL中のURLパス部に該当する部分の文字列（図4A参照）を利用した多様な情報処理が可能である。URLの表現能力の自由度が高くなり告知手段としての価値が高まる等の効果があるため、既存のWWWサーバに請求項1の機能拡張装置を備えたもの（以下、このようなWWWサーバを「FxURL対応WWWサーバ」と言う。）の産業上の利用価値は大幅に向上する。

【0035】

請求項2の機能拡張装置は、どのようなURLパス部の到来に対してエラーを誘発するのか（あるいは逆に誘発しないのか）やどのような時間帯にエラーを誘発するのか等の、エラーを誘発するための条件を変更する条件変更手段を備えている。

【0036】

これにより、状況に応じて既存のWWWサーバ処理手段と新規に設ける文字列処理手段とを目的に応じて切替利用したり、特定のURLホスト部を持つURLは常に既存のWWWサーバ処理手段により処理する等の条件を設定したり変更したりすることが可能となる。また、機能の異なる文字列処理手段を複数多種準備してこれらを切替利用したり連係させて利用する等も可能となり、より多様な情報処理が実現でき便利である。

【0037】

機能の異なる文字列処理手段を複数多種準備する際、共通部分の一本化や文字列処理手段自体の構造の簡素化は望ましい。このようにできれば、文字列処理手段の将来における機能拡張や維持管理が容易になる。

【0038】

請求項3の機能拡張装置は、機能拡張装置を構成するどの要素も（ただし請求項

2の条件変更手段を除く。)既存WWWサーバ処理手段の固定的部位を何ら変更することなく備える。ここで「固定的部位」とは、いったん使い始めたら

(1)変更することが容易ではない部位または

(2)通常は変更しないで利用する部位

のいずれかを言う。WWWサーバを構成するプログラム群で言えば、WWWサービスを実現しているプログラム自身(あるWWWサーバプログラムパッケージの場合は「httpd」という名前のバイナリファイルである。ソフトウェアのバイナリファイルではなく同等の機能を持ったハードウェアロジックである場合もこれに該当する。)がこれに該当する。

これに対して非「固定的部位」とは、例えば、WWWサーバの初期化処理部(608)が参照する初期化情報保持部(614)(あるWWWサーバプログラムパッケージの場合は「httpd.conf」という名前のテキストファイルである。ソフトウェアのテキストファイルではなくオン・オフ等が可能なスイッチ類や適用量を容易に変化させられるボリューム類もこれに該当する。)や、WWWサーバがHTTPレスポンス(図3参照)として送信するファイル保持部(620)の中のファイルを挙げるができる。

【0039】

つまり請求項3の機能拡張装置は、既存のWWWサーバの中心的部位をなんら変更する必要が無いため、容易にまたはコストをそれほど掛けずに適用できる。このため、一般にソースコードの入手が困難な商用のWWWサーバにも適用することができる。仮にソースコードの入手が困難でない場合でもソースコードを修正して再コンパイルする等の必要がないので速やかに適用することができる。また、仮にソースコードのコンパイルが不要なインタプリタ型の言語で「httpd」同等物が作られていた場合でも、ソースコードの修正に伴う万が一の機能劣化を回避することができる等のメリットがある。

【0040】

請求項4の機能拡張装置は、処理対象となるURLがWWWクライアント(典型的にはブラウザ)におけるURL入力表示欄(502)に入力された文字列に対応していることを第1の特徴としている。この発明ではこの文字列を「クライアント側入

力文字列」と言う。

【0041】

ここで「クライアント入力文字列」ではなく「クライアント側入力文字列」となっているのは、WWWサーバ(210)とWWWクライアント(202)との間にWWW中継処理手段(206)が介在する場合があるからである。

【0042】

WWW中継処理手段(206)が介在した場合、WWWサーバ(210)とHTTPにて通信するのは直接的にはWWW中継処理手段(206)であるが、WWW中継処理手段(206)は通常WWWクライアント(202)の要請に基づいてWWWサーバ(210)との通信を実施している。

【0043】

WWWサーバ(210)が処理対象とするURLがWWWクライアント(202)から直接送信されてきた場合もWWW中継処理手段(206)を経由して送信されてきた場合も、どちらもWWWクライアントからのデータであるという意味を込めて「クライアント『側』入力文字列」としたのである。

【0044】

第2の特徴として請求項4の機能拡張装置は、WWWクライアント(202)のURL入力表示欄(502)に入力した時の表示通りの文字列(このWWWクライアント側で入力された文字列を、以下「文字列1」とする。)をWWWサーバ(210)で再現するとしている。(WWWサーバ側で再現された文字列を、以下「文字列2」とする。)たとえネットワーク(208)中においてどのような表記形式で伝送されようとも文字列1と見た目上(つまり印刷した際に)同一の文字列をWWWサーバにて獲得する獲得手段を備えるとしている。

【0045】

これにより、WWWクライアント(202)のURL入力表示欄(502)に漢字等の全角文字や前記「Unsafe」な文字を含む文字列がURLとしてブラウザのURL入力表示欄(502)に入力された場合、すなわち、前記FxURLがブラウザのURL入力表示欄(502)に入力された場合でも、

(1)両WWW処理手段((202)と(210)のこと。)が採用する漢字コード体

系の相違、

(2) 2以上のWWWクライアント(202)間のエンコード処理方法の相違、または

(3) WWWクライアント(202)におけるエンコード実施の有無の相違

によらず、前記文字列処理手段はWWWサーバが備える記憶手段の情報を利用しつつWWWクライアント(202)のURL入力表示欄(502)に入力した時の表示通りの文字列1を文字列2として得ることができる。

【0046】

ただし、URL入力表示欄(502)に(エンコードしないままの)「#」そのものを含む

`http://www.ox.com/index.html#sales`

というような文字列を入力した場合、その「#」以降の文字列(「#sales」)は通常はWWWサーバ側(WWWサーバまたはWWW中継処理手段)には伝達されないため、その「#」以降の文字列をWWWサーバで再現することはできない、つまり文字列1と文字列2が同一にはならない;としている。「#」以降の文字列がWWWサーバ側には伝達されるのであれば、文字列1と文字列2は同一にすることができる。

【0047】

前記FxURL(またはそのサブセットであるKURL)がブラウザのURL入力表示欄(502)に入力できることにより、図10の(1001)、(1007)、(1009)に示すような漢字等を含むKURLを用いることが可能となり、URLの表現手段としての価値は飛躍的に向上する。また図12Cのように、前記「『Reserved』である文字群(1106)」の文字を含むごくふつうの英文表記もFxURLのURLパス部に置くことが可能となる。両者の混合、すなわち、FxURLのURLパス部に「Reserved」である文字群(1106)の文字と漢字等の文字の両方を含む文字列を置くことも可能である。

【0048】

資料2(出典は後述)の新聞の記事によれば、

「英語でURLというアドレスを入力してホームページを探すのは日本人には骨が折れる」

というある人物の発言が紹介されていたり

「インターネットの窓口にもURLの日本語変換ソフト（中略）がほしいという声は少なくない」

という認識が示されていたりする。これがインターネット関連業界またはパソコン利用者の標準的認識と考えてほぼ良いであろう。上述したように、請求項4の機能拡張装置はこれに十分応えることができると発明者は考えている。

【0049】

なお、請求項4の機能拡張装置を作る際には、この機能拡張装置内部で用いる漢字コード体系を処理の早い段階で統一することが望ましい。より複雑な処理をこの機能拡張装置の後段で実施する際に漢字コード体系の相違を意識せずに処理を進められることはメリットになる。

【0050】

請求項5の機能拡張装置は、処理対象のURLがブラウザのURL入力表示欄（502）以外から与えられる文字列に由来していることを第1の特徴としている。第2の特徴については請求項4の機能拡張装置と同様である。以下、第1の特徴についてもう少し詳しく説明する。

【0051】

一般的なWWWサービスの実施の形態においては、WWWクライアント（202）はブラウザというプログラムを備えたパソコンであることが多い。ブラウザの画面インターフェースの概要は図5に示すようなものである。ブラウザのURL入力表示欄（502）からWWWサーバにおいて処理対象となるURL——それはTrURLかもしれないしFxURLかもしれない——が与えられるのである。つまり、WWWサーバにおいて処理対象となるURLの入力または選択は、人間がキーボードを用いてキー入力する場合でもいくつかの項目の中からマウスを用いて選択する場合等でも、手動で行われる。ある者は商用ブラウザが備える良好なGUI（グラフィカルユーザインターフェース）によりこれを行うであろうし、別のある者はGUIではなくCUI

（キャラクタユーザインターフェース）を備えたブラウザによりこれを行うであろう。画面の体裁はともかく、WWWクライアントとして特化されたプログラム（すなわち「ブラウザ」）を利用することであろう。

【0052】

請求項5の機能拡張装置はそうではなく、

(1)手動でURLを入力するもののブラウザほど良好な（または専用の）GUIやCUIを持たないプログラムまたは

(2)所望のURLを自動で生成して自動でWWWサーバへと送信するプログラム等が利用されることを想定している。

【0053】

前者の例として「telnet」と称するプログラムを挙げることができる。後者の例は、たとえばHTTPを処理することができる自作プログラムを挙げることができる。後者のプログラムは人手を介することなく処理が進められるため、自動的かつ大量の処理を一括でまたは適宜分散させて処理するのに好適である。同時に、そのようなプログラムにはGUIはもとより原始的なユーザインターフェースさえ持たせる必要が無いいためソースコード（または実行ファイル）の規模は小さくなり、高性能でないパソコンでも好適な処理速度が得られるだろう。なお、自動的な処理を行うための条件設定に際しては人手を介する場合もあろう。

【0054】

このような特徴をもった上記(1)または(2)記載のプログラムを用いることにより生成されたURLをWWWサーバにおいて処理対象とするのが請求項4の機能拡張装置である。従って、WWWサーバにおいて処理対象となるURLは必ずしも人手を介したものである必要は無いので、自動的かつ大量の処理を一括でまたは適宜分散させて処理するのに好適である。

【0055】

しかも、URLとなる文字列をブラウザにおけるURL入力表示欄（502）から与える必要がないため、URL入力表示欄（502）からは入力できないまたは入力が困難な文字列でもWWWサーバにおいて処理対象となるURLとすることができる。これにより、例えば入力困難な漢字や記号類を用いたより多様な表記の文字列——それがFxURLである——をURLをして用いることができ、便利である。

【0056】

請求項6の機能拡張装置は、

(1)URLパス部に漢字、ひらがなまたはカタカナが含まれており、かつ、

(2) CGI プログラム等へのパラメータ伝達を指示すると RFC にて定義された文字列（現バージョンの RFC においては「?」）を RFC の定義通りに用いる用途では含まないこと；

を特徴としている。逆に言えば、漢字等を含む URL パス部において CGI プログラム等へのパラメータ伝達のために「?」が使われている場合、すなわち、URL パス部が RFC に記載されている URL の通常の表示様式に準拠した文字列である場合、その URL が処理可能であることはこの発明の範囲ではない；ということである。一例を挙げれば、URL パス部が「?イロハ」あるいは「index.cgi?イロハ」なる文字列である場合 URL についてその URL が処理可能であることはこの請求項における発明の範囲ではない。

【0057】

もう少し具体的に説明する。

図 13 A または図 13 C に示す URL が処理できるようになることはこの発明の範囲であるが、図 13 B に示す URL が処理できるようになることはこの発明の範囲ではない。

なぜなら、現在市場で主流となっているブラウザを用いて発明者が実験した範囲では、既存の WWW サーバが通常備えている CGI 処理手段のパラメータとして図 13 B に示す URL の URL パス部を処理可能なようにブラウザが適切にエンコードすることになっているからである。

（ただし、前記図 13 B に示す漢字を含む URL を一見「適切に」処理するような実装は RFC の規約に違反しているかもしれないし、発明者が実験した範囲でたまたまうまく動作しただけかもしれない。）

よって図 13 B に示す URL が処理可能になることは新規性が無い。それに、既存の WWW サーバが図 13 B に示す URL を処理するに際しては請求項 1 に記載したような意図的にエラーを誘発するようなエラー誘発手段を用いていない——すなわち

、正常処理の範疇として情報処理される——ため、この意味からも図 13 B に示す URL が処理可能になることはこの発明の範囲ではない。

【0058】

URL を情報表現手段（たとえば広告媒体などの告知媒体）として捉えた場合、次

の理由により図13Aに対して図13Bは劣っていると言って良いであろう。URLを読む側が基礎的な英語力を有していると仮定する場合、図13Aについては、URLホスト部とURLパス部の部分に着目して「My name is 林毅」、すなわち、「私の名前は林毅です」という意味を想像（意味に解釈）することができよう。図13Bもそのように想像・解釈できなくはないかもしれないが、URLパス部の先頭にある「?」は余計である。通常の文章表現においてこのような「?」を用いることは無いし、個人や法人等の名前自体に「?」が含まれることも無いであろう。

【0059】

図13Cに示す例は、「あなたの名前は林毅ですか?」という意味を想像（意味に解釈）することができよう。図13Bとは違い（URLパス部の末尾にある）「?」は余計ではない。文の疑問符としての役割を果たしていると解することができるからである。

【0060】

なお図13Bの表記は、「?」の前に存在すべきCGIプログラムのファイル名等が省略されたものである。これを省略せずに表記すれば、たとえば、図13B2のようになる。図13B2のURLは告知媒体としての価値は大幅に低下していると言って良いであろう。

【0061】

請求項7の機能拡張装置は、URLパス部が日本語、英語、フランス語のような国または地域で使われている言語における語句（単語）を含んでいることを特徴としている。このことは自動的に、句または文を含むことを意味する（図14のA、B、C、DおよびFを参照）。エスペラント語のような人工言語を排除しない。

【0062】

ブラウザにおけるURL入力表示欄（502）からまたはその他の手段により、

- (1) ISO 8859-1文字コード集合中の非ASCII文字コードの文字や
- (2) いわゆる外字（既存の漢字コード体系において標準的なコードが割り当てられておらず、利用者が個々にコードを割り当てている文字のこと。）として扱っ

ている文字もしくは

(3)それらを混合した文字列

をURLとして入力または指定等できるのであれば、これを用いることにより一層多様な表現が可能となる。図14Hはフランス語の句が含まれている(1)の例、図14Iは「高」の旧字体が含まれている(2)の例である。この説明では「ISO 8859-1」を取り上げたが、「ISO 8859-2」や「ISO 10646」やその他の文字コード集合についても同様に考えることができる。

【0063】

請求項7の機能拡張装置はまた、特定の学術分野（例えば数学）において用いられる式などの表現形態を含んだり、コンピュータ言語の命令等を含むことを特徴とする場合がある。図14Gは数式（ x の2次方程式）を含むURLの例である。図14Eはコンピュータ言語の一種であるC言語を含むURLの例である。

【0064】

これにより、多様な言語の文や式などを含む多様な表現力を備えたURLを用いることができるので、表現力豊かなURLの利用が可能になる。URLの情報表現手段としての利用価値は一層向上し、営業活動等への利用に好適である。消費者の購買意欲をそそるようなワクワクするような表現をURLに用いることも可能であろう；と発明者は考える。

【0065】

請求項7を突き詰めて考えれば、任意の表現手段を採りうることが可能だと分かる。言語系としては点字や手話、あるいはジェスチャーやボディランゲージであっても良いかもしれない。

ただし、手話やジェスチャーのような動作をブラウザのURL入力表示欄（502）に表現しようとするれば「アニメーションGIF」（画像データを用いた連続表示したスライドのようなもの）を用いれば可能かもしれないが表現内容が時間とともに変化してしまう等の理由から実装（実現）は難しい。しかし、点字ならばURL入力表示欄（502）の実装方法を工夫することにより入力や表示が可能であろう。

このような考えを推し進めれば、任意の画像データをURL入力表示欄（502）

にて取り扱うことが可能であるとの推察が付く。上述した理由から以下では静止画に限って話しを進める。

【0066】

これを実現するためには、URL入力表示欄（502）を既存のブラウザにおいて実装されているようなく所定の文字列の表示、入力、編集が可能な領域＞と捉えるのではなく、＜任意の画像データの表示、入力、編集が可能な領域＞と捉え直せば良い。

具体的には、現在のような＜1行のみのテキストエディタ＞ではなく＜横長の長方形領域を編集範囲とする画像エディタ＞として新たな仕様でURL入力表示欄を実装し直せば良い。もちろん通常の画像エディタと同様、文字列の表示、入力、編集も可能とするのである。

なおこの「新たな」「URL入力表示欄」（以下、これを「画像対応型URL入力表示欄」と言う。）は、外見上は従来のURL入力表示欄と同一にすることもできるので、図5に示すURL入力表示欄（502）は画像対応型URL入力表示欄であると見做すこともできる。

【0067】

ここまで考えを発展させると、上述のようにURLパス部に画像データ等を含むURL（以下、このようなURLをビジュアルURL、略して「VURL」と言う。）はRFCで定義されたURLの範疇を越えてしまうように思える。しかしながら、例えば、点字や画像データをRFCで定義されているMIME (multipurpose internet mail extensions) という規格のBASE64形式にてエンコードするという規約をVURLをやり取りする双方が合意すれば、点字や画像データをURLパス名部に表現することが可能となる。

【0068】

この場合、

-
- (1) WWWクライアントにおけるURL入力表示欄における表現物（表現物1）と表現物1を受信したWWWサーバにおけるデコード後の表現物（表現物2）については点字や画像データという形式で情報が表現される一方、
 - (2) WWWクライアントとWWWサーバとの間にあるネットワーク中ではBASE64形式の

文字列として表現されたもの（表現物3）とする；

というような実施形態を想定することができる。VURLの例を図14Jに示す。URLパス部には人の顔に似せた小さなアイコンのような画像が置かれている。

【0069】

このような機能拡張も、この発明の請求項1に記載されているような「WWWサーバがURLを処理するに際して意図的にエラーを誘発するエラー誘発手段」や「少なくともURLパス部を利用して処理を行う文字列処理手段」等を備えるこの発明の機能拡張装置を用いることにより、WWWサーバ側については比較的容易に対応が可能である。

なお、WWWクライアント（ブラウザ）側については、「プラグイン」と呼ばれている良く使われている機能拡張手段により対応が可能である。

【0070】

なお、VURLの画像データ部分をBASE64でエンコードした場合、画像データは

「a」～「z」、

「A」～「Z」、

「0」、「1」～「9」、

「+」、「/」、「=」

の合計65種類の文字の組合せで表現されることになるが、FxURLならば文字「/」を始め65すべての文字について何ら制約を受けずにURLパス部に記述可能である。

【0071】

VURLによれば点字（正確には、点字風の画像データ。）や絵文字を含む任意の画像データを情報表現手段としてURL中に用いることができるため、例えば、

(1)通常の日本語や英語等による表記を理解できない人々、

(2)点字や絵文字の方が速読できる（またはなじみが深い）人々、

(3)絵文字を用いた簡便な意志伝達を希望する人々

などにとっては有意義なものとなろう。点字を含むVURLは身障者の人々がインターネットを活用する際の一助となるかもしれない。ただし、平面的に印刷された点字ではなく凹凸のある真の点字の入力や読み取りに際しては別途専用の装置等

が必要になる。

【0072】

なお、VURLの表記内容を読み取ることは容易であろうが、VURLが紙などの印刷媒体に表示されているような場合は（文字列と違って）読み取った表記内容と同一のものをURL入力表示欄に入力することは多くの場合困難であろう。この場合、例えば、

- (1)VURLを印刷した近傍にそのVURLに対応する2次元コードを添え、その2次元コードを2次元コードリーダーを用いて機械的に入力する；
 - (2)VURLがカードに印刷されているならば、カードに磁気ストライプ等を設けてその中にエンコードしたVURLを記録し、磁気カードリーダー等を用いて機械的に入力する；
- などの方法のより対処可能となる場合もあろう。

【0073】

更に、URL入力表示欄（502）に入力される文字列を図形情報と捉えることにより、用いる文字フォントの違い（例：明朝体とゴシック体）により実際に意味するURLの解釈を変えるという実装を行うことも可能である。この場合、

- (1)従来の文字列（テキスト）としての情報にフォントの種別についての情報を付帯させてWWWサーバに伝達するか、または、
- (2)入力されたURLをそのまま画像データとして捉えて上述の方法でWWWサーバに伝達する

等の方法が想定可能である。

【0074】

請求項7の機能拡張装置はまた、URLパス部に暗号化されたデータを含むこともできる。データを暗号化してこれを文字列として出力した場合、その文字列は通常にASCII文字コード集合の図形文字を用いることになるが、FxURLにはTrURLの

ような「Unsafe」または「Reserved」となっている文字種は無いので、特別なエンコード処理をすること無く暗号化されたデータをURLパス部に置くことができる。図14Kに例を示す。「encrypted/」以降に示すのは、あるデータを（コンピュータ業界において）「DES」と呼ばれている暗号方式により暗号化した文字

列である。これはあくまでも一例であり、暗号化方式は任意である。

【0075】

請求項8の機能拡張装置は、URLパス部に検索を指示する文字列が含まれていることを特徴とする。「検索を指示する文字列」には大きく分けて2種類ある。記号を利用するものと自然言語によるものである。

【0076】

第1の記号を用いた検索指示では、コンピュータ業界において「正規表現」や「ワイルドカード」と呼ばれる表現方法を用いる。これらを用いれば、図15のA、B、C、DおよびEのようなURLが利用可能となる。

【0077】

図15Bはワイルドカードを用いた例である。WWWサーバはファイル保持部(620)のディレクトリ領域(図8A参照)を検索して、「a」で始まり「z.html」で終わるファイル名(例:「abcdz.html」、「aあいうz.html」、「a+-=z.html」)を持つファイルを探し出す。

図15Aもワイルドカードを用いた例であり、「a」で始まり「z.html」で終わる8文字のファイル名(例:「abz.html」、「a-z.html」、「a.z.html」)を持つファイルを探し出す。該当するファイルが2以上ある場合は、

- (1)最初のファイルを検索結果としたり
- (2)検索結果を順にすべて処理する

等のルールを別途定めて実施することになる。また、上記の例において全角を文字を含むファイル名を検索の対象にすべきか否か等については別途ルールが必要であろう。

【0078】

図15Cは正規表現を用いた例である。ファイル名が数字のみから成るファイルを探し出すよう指示している；と解釈できる例である。

【0079】

図15Dは、WWWサーバ処理手段の外部にあるデータベース(この例では当該データベースの識別子は「names.jp」としている。日本の人名データベースの意に解釈されたい。)から名前の最初が「田中」である人物を探し出すよう指示して

いる；と解釈できる例である。

【0080】

同様に図 15 E は、データベース「names.jp」から名前の末尾が「子」である人物を探し出すよう指示している；と解釈できる例である。

【0081】

実際の運用に際しては、どのような表記（文字列）がどのような意味を持つのかについて、事前に定義したり動的に決定したりするなど、WWWクライアント（202）側とWWWサーバ（210）側とで一定のルールに合意する必要がある。

【0082】

第2の自然言語を用いた検索指示について説明する。

【0083】

図 15 E 2 は、図 15 E と等価な意味をワイルドカードを使うのではなく自然言語を用いて表現した例である。日本語以外で記述しても良い。また、図 15 F（またはデータベース識別子を省略して図 15 F 2）のように質問文を含むこともできる。このようなFxURLをWWWクライアント（202）側で用いてWWWサーバ（210）側にて処理しようとするれば、

(1)URLパス部の部分を言語処理して意味を獲得する言語処理手段を別途備えるか、または、

(2)WWWサーバ（210）側で解釈可能な構文のパターンを予め定めておき、WWWクライアント（202）側ではそのパターンに従って文を組み立ててURLパス部に含む；等の処理が必要になろう。後者の方法は、書式がきちりと定まっているという意味においてコンピュータ言語のプログラミングの要領に似ている。

【0084】

以上に述べたように、

(1)複数のファイルを一度の要求で獲得可能な検索式を成す文字列をURLに含んだ
り、

(2)自然言語の質問文を成す文字列をURLに含むこと

等が可能となり、便利である。この技術を用いることにより、インターネットにおける新しいタイプの検索サービスを提供することが可能となろう。

【0085】

請求項9の機能拡張装置は、下記に示す2つの対照的な手段のうちの少なくとも一方を備えることを特徴とする。

【0086】

その第1の手段である「特別文字列普通解釈手段」により、HTTPの規約により特別な意味が与えられている「?」や「/」等の「Unsafe」である文字群(1104)や「Reserved」である文字群(1106)の文字についてその文字本来の用途に用いることが可能となる。

【0087】

これにより、「4分の3」という分数をURLパス部に表記できる(図16A参照)。

通常ならば図16Aは、<「answer=3」という名称のディレクトリの下にある「4=0.75」という名称のファイルを指示するもの>と解釈すべきURLである。なぜなら、URLの様式を定めたRFCにより「/」は階層構造の境界を示すとされているからである。

TrURLではこのような解釈になるが、FxURLでは必ずしもこのように解釈しなくてもよい。このように請求項9の機能拡張装置は、「/」を階層構造の境界を示す記号以外の意味に用いることを可能にする。「/」以外も同様である。

【0088】

このように「特別文字列普通解釈手段」により、より自然な表現の文をURLパス部に含むことが可能となる等のメリットが得られる。

【0089】

その第2の手段である「普通文字列特別解釈手段」は、上記「特別文字列普通解釈手段」とは逆に、何らかの文字列を階層構造の境界を示すものとして解釈することを可能にする。例えば、図16BのKURLには階層構造の境界を示すものは無い。これを、「の中の」または「の」を「/」と等価であると解釈することにより、図16B2のKURLに置換することができる。両KURLの下線部に注目されたい。

【0090】

これにより、ファイル保持部（620）におけるアクセスすべきファイルの所在が確定し、目的のファイルを読み出すことができるようになる。

【0091】

図16B2のKURLは漢字等を用いた日本語の文となっているが、この例における図16B2から図16B3への変換処理については、URLパス部の内容を言語的意味を理解して処理するのではなく、単なる置換処理で十分である。（ただし、「の中の」から「/」への置換処理を「の」から「/」への置換処理に先行して実施する程度の工夫は必要である。）このため、（文法解釈等を伴う言語処理を行う場合を考えれば）この変換処理に要する作業量はそう多くはない。

【0092】

TrURLでは「の」をこのような意味で用いるという用法はないし、漢字等も使うことができないため、図16B3のように表記せざるを得なかった。図16Bに示すような表記方法（つまりKURLもしくはこれを包含するFxURL）が普及すれば、コンピュータ利用の「参入障壁」を数段下げることができる可能性があり、WWWサービスの利用者の増大等が期待できる。併せてインターネットのサービス全体の利用者増も期待できる。

【0093】

なお、階層構造の境界を示すものとして解釈させる「何らかの文字列」はローマ字表記の日本語の句でもよいし、ローマ字表記の英語の句でもよいし、その他記号列等でもよい。図16B4に示すのはローマ字表記の英文である。この場合、まず「△of△」（ここで「△」はASCII文字コード集合における空白文字。C言語の16進数表記で0x20。いわゆる半角スペースを示す。図面においても同様。）を「/」に置換した後、更に左右の語の並びを入れ替えることにより図16B3を得ることができる。更に英和翻訳等を行えば図16B2を得ることもできる。

【0094】

上記の説明ではRFCにより「階層構造の境界を示すもの」と定義付けられた事項とそれに対応する文字列を取り上げたが、これ以外の事項についても同様である。

【0095】

このように「普通文字列特別解釈手段」により、より自然な表現の文をURLパス部に含むことが可能となる等のメリットが得られる。

【0096】

以上のように、請求項1を頂点とするこの発明は、応用範囲が極めて広い。これは、前記文字列処理手段における情報処理の内容を特段制限する事由が無いことによる点が大きく寄与している。特に請求項3の発明によれば既存のWWWサーバの固定的部位は何ら変更する必要がないためこの機能拡張装置の利用は容易であり、これがこの発明の価値を高めることになる。

【0097】

この発明による機能拡張装置を利用するためには、対象となる装置が

(1)例外事象発生時の処理手段を呼び出す仕組みと

(2)その処理手段として所望のものを設定可能

という2の条件を満たしていれば良い。たとえば「CPU」は「WWWサーバ」ではないが、CPUはこれらの条件を満たしているものが多い。

【0098】

このことを熟考すれば、請求項1が機能拡張の対象としている装置をWWWサーバにのみ限定していることについて疑問が生じて来る。「この限定を取り払うとどうということになるのか？」発明者は自問自答した。

【0099】

コンピュータのファイル等のデータを「資源」と捉えるならば、URLはインターネットにおける資源の同定用文字列と言える。実際、「URL」の正式名称である「Uniform Resource Locators」がそれを示している。そして、資源の所在を指し示す文字列を「アドレッシング用文字列」と呼び直してみると、

(1)URLと並んでインターネットにおいて頻繁に利用されている文字列である「電子メールアドレス」や

(2)URLと電子メールアドレスに共通の要素である「ドメイン名」

等も「アドレッシング用文字列」と言って良いだろう。

【0100】

ここまで考えを進めるとこの発明は、アドレッシング用文字列を処理するサーバ

全般に適用できそうだと推測が付く。具体的には、

(1)電子メールの配送等を行うための手段である電子メールサーバや、
(2)ドメイン名からIPアドレスを得るための（またはその逆の用途のための）手段であるDNS (Domain name system)サーバにも適用できそうなのが推測できる。FTP (file transfer protocol)サーバにも適用できるであろう。すなわち、電子メールサーバ（「sendmail」という名称のプログラムが有名である。）を例に採れば、次のような請求項を設けることが可能である。

【0101】

「既存電子メールサーバが電子メールのアドレスを処理するに際して意図的にエラーを誘発するエラー誘発手段と、前記既存電子メールサーバが記憶手段に記憶させたデータのうち少なくとも電子メールアドレスのユーザID部を利用して処理を行う文字列処理手段と、前記エラー誘発によりエラーが発生した時に前記既存電子メールサーバから前記文字列処理手段へと処理を分岐させる分岐手段とから成る機能拡張装置。」

【0102】

更に考えを推し進めれば、この発明は、インターネット以外の領域にも適用できそうなのが推測できる。例えば、電話通信網における電話番号に基づく回線交換システムにも適用可能であろう。

【0103】

このように考えるうちに、これらの適用可能な用途をすべて個別に列記して請求項として記述する方法では限界があると発明者は考えた。そこでこの発明の本質を抽出し、新たな請求項とした。それが請求項10と請求項11である。

【0104】

請求項12の発明は、請求項1の発明を方法の発明としてクレームしたものである。すなわち、WWWサーバ（これはコンピュータ装置を用いたプログラムと解釈してもよいし所定の機能を果たす専用ハードウェアと解釈しても良い。）がURLを処理するに際して次の3の手段により機能拡張するという発明である。

【0105】

第一は意図的にエラーを誘発するエラー誘発設定ステップを備えること。

第二はWWWサーバが記憶部に保持したデータのうち少なくともURLパス部を利用して処理を行う文字列処理ステップを備えること。

第三は上記のエラー誘発設定ステップが原因となってエラーが発生した時にWWWサーバ内部のルーチンから新たに設ける文字列処理ルーチンへと処理を分岐させる分岐処理ステップを備えること。

【0106】

請求項13の発明は、請求項1の発明を媒体の発明としてクレームしたものである。すなわち、請求項12の発明の範疇にあるWWWサーバのうちコンピュータのプログラムとして提供される「機能拡張プログラム」を記録した記録媒体である。

【0107】

容易に想像可能であろうが、請求項1以外の機能拡張装置の発明についてもこれに対応する方法の発明と媒体の発明が存在する。

【0108】

発明者が更に考えを推し進めたところ、エラー誘発手段は必ずしも機能拡張装置の構成要素として含まれる必要はないであろうとの知見を得たため、請求項14としてこの発明を記述した。また、請求項14における「既存情報処理手段」をWWWサーバに限定した場合の発明を請求項15として記述した。

【0109】

この発明において「文字列」とは、文字が1以上連続したものを言う。ここで「文字」とはコンピュータ上で表現できる任意の文字を言う。別の言葉で言えば、何らかの文字コード集合（コード体系）において文字コードが一意に定まっているものである。

【0110】

「文字コード集合」とは、対応するコードが一意に定められている文字の集まりもしくはその集まりに対して付けられた名称を言う。

【0111】

「全角文字」とは、日本語等の文章表記用としてJISで定められた漢字、ひらがな、カタカナ等を含む文字コード集合として「JIS X 0208」として定められた文

字群の各文字を言う。1文字を1バイト以内（7ビットまたは8ビット）で表す文字を通称「半角文字」と呼んでいるのに対して1文字を2バイト（16ビット）で表現するため「全角文字」と呼ばれているようである。1文字を2バイト以上で表す通称「補助漢字」（JIS X 0212）や、中国、韓国、台湾で用いられている漢字等で1文字を2バイト以上で表すものを含む概念である。

【0112】

文字の「集合」としては全部または大部分が同一であっても対応するコードが異なる場合がある。全角文字のうち「JIS X 0208」について言えば、「JIS」、「EUC」、「シフトJIS」（いずれも通称または俗称。）の3種が場面に応じて用いられている。このような場合、「3種の漢字コード（漢字コード体系）が存在する」などと言う。国際標準化機構が定めた「ユニコード」（ISO 10646）の漢字等の部分も一部重複する。

【0113】

「文字列処理」とは、文字列を用いた情報処理のことを言う。文字列を複数の文字列に分解したり、日本語を成す文字列を英語に翻訳したり、何らかのルールに基づいてある文字列を別の文字列に置換したり、文字列の文字数のカウントするなどを例として挙げることができる。URLに関連して言えば、URLをURLホスト部とURLパス部とその他の部分とに分解するのは文字列処理の一例である。

【0114】

この発明において「プログラムを記録した記録媒体」とは、

フレキシブルディスク、

CD-ROM、

光磁気ディスク、

ハードディスク、

メモ리카ード、

ICカード、

ROM、

パンチカード、

テープ

等を含む概念である。また、コンピュータによって直接実行可能なプログラムを記録した記録媒体だけでなく、いったん他の記録媒体（ハードディスク等）にインストールすることによって実行可能となるようなプログラムを記録した記録媒体や、暗号化されたまたは圧縮されたプログラムを記録した記録媒体も含む概念である。

【0115】

「CGI処理手段」または「CGIプログラム」とは、WWWサーバ処理手段からCGI (common gateway interface)なる機構により呼び出される情報処理手段のことである。とりわけ「CGIプログラム」と言った場合は、その情報処理手段がプログラムとして実装されていること示す。単に「CGI」と言う場合もある。

【0116】

「トップページ」とは、通常、ホームページの中で最初に表示されるページである（図9参照）。本来「ホームページ」とは、今日で言うホームページ（以前は「Webサイト」等と称していた。）の入口のページを指していたが、誤用が広がって定着したものである。ホームページには複数の文書ファイルや画像データ、あるいはCGIプログラムが関連しており、それぞれのファイルは独自のアドレス（URL）を持つ。一般に「ホームページのアドレス（もしくはURL）」と言った場合には「トップページ」のURLを指している。

【0117】

「WWWサーバ」とは、WWWサービスのサーバ側の機能（図3参照）を提供するプログラムである。そのようなプログラムが機能するに際して必要なファイル等（図6参照）も含む概念である。そのようなプログラム（および前記必要なファイル等）がワークステーションやパソコン等のコンピュータにインストールされた装置（図17参照）を含む概念である。WWWサーバの機能がソフトウェアまたはハードウェアとして内蔵された専用の装置も含む概念である。とりわけ、上記プログラムや装置が動作中のものをこう呼ぶ。また、WWWサーバ処理装置上で動作するプログラムに着目している場合は「WWWサーバプログラム」と言う。この発明による機能拡張を致したことによりFxURLの処理が可能となったWWWサーバを含む概念であるが、特にこの機能拡張したWWWサーバを示したい場合は「FxURL対応WW

Wサーバ」と言う。

【0118】

「WWWクライアント」とは、WWWサービスのクライアント側の機能を提供するプログラムである。そのようなプログラムが機能するに際して必要なファイル等も含む概念である。そのようなプログラム（および前記必要なファイル等）がワークステーションやパソコン等のコンピュータにインストールされた装置を含む概念である。WWWクライアントの機能がソフトウェアまたはハードウェアとして内蔵された専用の装置も含む概念である。とりわけ、上記プログラムや装置が動作中のものをこう呼ぶ。また、WWWクライアントの画面インターフェース（図5参照）に着目している場合は「ブラウザ」と言う。

【0119】

「ネットワーク」とは、広域の公開ネットワークであるインターネット、組織内のネットワークであるイントラネットならびに比較的小規模なネットワークであるLAN (local area network)を含む概念である。通信規約にTCP/IP (RFCで定められているインターネット用の通信規約)を用いないネットワークを含むことを妨げない。

【0120】

「ディレクトリ」とは、ファイルシステム（ファイルを管理する手段またはファイルを格納した媒体のこと。）においてファイルのインデックス情報が格納された特殊な情報領域である。OSによってはこの情報領域がファイルの一種として実装されている場合がある。

【0121】

「ファイル」とは、データファイルとプログラムファイルと特殊ファイルの総称である。ここでデータファイルとは、テキスト（文字列、文書）情報や画像データ、音声データ等のデータを格納したファイルである。プログラムファイルは「プログラム」または「実行ファイル」とも呼ばれ、コンピュータ上で実行可能であるファイルである。特殊ファイルの代表例は前記ディレクトリである。ディレクトリ以外には、OSによっては、画面やキーボードをファイルとして扱うものもある。なお、単に「ファイル」と言った場合、狭義にはデータファイルとプログ

ラムファイルを、更に狭義にはデータファイルのみを指す。意味が不明確になり
そうな場面やファイルの内容を限定したい場合は、「データファイル」、「プロ
グラムファイル」または「ディレクトリ」と表記する。

【0122】

この発明において「OS」とは、OSの核である「カーネル」と呼ばれる部分に加え
、「ライブラリ」と呼ばれるカーネルの周辺モジュール、「シェル」等と呼ばれ
る対ユーザ対話機能提供モジュールにエディタや文字列処理プログラム等のカー
ネルに付随するツール等と称されるファイルを含む概念である。

【0123】

この発明における「資料1」の出典は、「日経コミュニケーション」（日経BP社
発行）の1998年4月20日号の第111ページである。

【0124】

この発明における「資料2」の出典は、1998年4月15日付け「日経産業新聞」第
6面である。

【0125】

【発明の実施の形態】

（実施形態1）

図2にこの発明によるFxURL対応WWWサーバを用いたWWWサービスの一実施形態を
示す。

【0126】

LAN（204）に接続されたWWWクライアント（202）はWWW中継処理手段（2
06）とインターネット（208）を介してWWWサーバ（210）と通信可能な
状態にある。インターネット（208）に接続されたWWWクライアント（202
）はインターネット（208）を介してWWWサーバ（210）と通信可能な状態
にある。

【0127】

WWWサーバ（210）には既存のWWWサーバとFxURL対応WWWサーバとがある。機能
拡張した部分以外は共通なので、機能拡張した部分以外は特に分けずに記述する
。必要に応じて「既存の」または「FxURL対応」なる形容詞を前置する。よって

、図2に示すWWWサーバ(210)は、既存のWWWサーバである場合とFxURL対応WWWサーバである場合とがある。ただし、図2の中の少なくとも1のWWWサーバ(210)はFxURL対応WWWサーバである。

【0128】

WWWサーバ(210)の構成要素および関連データの関係を示すブロック図6と図7に示す。図6は既存のWWWサーバのものである。図7はFxURL対応WWWサーバのものである。

【0129】

WWWサーバは、自分自身の初期設定を行うために初期化処理部(608)を実行したのち、WWWクライアントからのHTTPリクエスト待ちの状態となる。この時各処理部と全体の制御を行うのが制御部(610)である。初期設定に際しては初期化情報保持部(614)に与えられた動作条件等を読み込み、そのデータを必要に応じて変数情報保持部(618)に格納する。この結果変数情報保持部(618)には、HTTPレスポンスとしてWWWクライアントに送信すべきファイル(図3B、図8参照)等を格納したファイル保持部(620)の所在等が格納される。

【0130】

データ受信部(602)はWWWクライアントからのHTTPリクエスト(図3A参照)を文字列として取り込むためのものである。取り込んだURL等の文字列データを必要に応じて変数情報保持部(618)に格納する。

【0131】

制御部(1608)はURLおよび関連情報のデータをURL解釈部(606)に渡す。URL解釈部(606)はURLパス部を取り出す等の目的で、図18に示すURL分解手段(1804)により受け取ったデータのうちURL(1802)をURLプロトコル部(1806)、

URLホスト部(1808)、

URLポート部(1810)、

URLパス部(1812)

とに分解する(図4参照)。

【0132】

時としてURLパス部（1812）が省略されているURLがある。また、多くの場合URLポート部（1810）は省略されているが、この場合はHTTPの規約に沿ってポート番号80が仮定される。多くの場合URLプロトコル部（1806）は「http」であるが、場合によっては「https」となる。これはセキュリティを強化したHTTPである。URLプロトコル部（1806）が「ftp」や「wais」等となる場合があるが、この実施の形態ではこれらの扱いは言及しない。

【0133】

なお、ブラウザのURL入力表示欄（502）においてURLを入力する際にURLホスト部以降を入力した場合に「http://」等の文字列を自動的に前置するブラウザがある。このような前置が行われた場合、この発明においては、当該文字列はブラウザの利用者が入力したものであると見做すことを妨げないものとする。

【0134】

このようにして獲得したURLパス部の文字列は、ファイル保持部（620）にアクセスする際の情報となる。ここで「アクセス」とは、多くの場合、

- (1)特定のデータファイルの内容を読み出すことを求めるものであるか、
 - (2)特定のプログラムファイルを実行することを求めるものであるか
- である。

【0135】

特定のファイルにアクセスする際にWWWサーバは、初期化処理部（608）により指定されて変数情報保持部に保持されている「ドキュメントルート」なる変数（以降「DocumentRoot」と表記する。）と階層構造の境界を示す文字「/」とをURLパス部に前置し、実際にアクセスするファイルの所在を決定する。この様子を図19に示す。

【0136】

また、「スクリプトエイリアス」なる変数（以降「ScriptAlias」と表記する。）が変数情報保持部に保持されていれば、CGIプログラムの所在の決定に際してScriptAliasの設定内容に沿った文字列の置換が行なって実際にアクセスするファイルの所在を決定する。

【0137】

また、「ディレクトリインデックス」なる変数（以降「DirectoryIndex」と表記する。）が変数情報保持部に保持されていれば、ファイル名の自動補完を行う。この実施の形態においては

index.html

となっている。

【0138】

URLパス部の文字列が階層構造の境界を示す文字「/」で終わっている場合、DirectoryIndexが設定されていない場合は実際にアクセスするファイルはディレクトリであると判断して当該ディレクトリの内容、すなわち、図8Aに示すようなインデックス領域の情報自体を表示する。一方DirectoryIndexが設定されていた場合は、変数情報保持部に保持されているファイル名を前記「/」に後置し、それを実際にアクセスするファイルの所在（これを「フルパス」と言う。（1910）、（1914）参照。）とする。

【0139】

DirectoryIndexに関する上記仕組みがあり、また、多くのWWWサーバではこの設定がなされているため、通常、図19に示すように、

http://www.any-domain.or.jp/

なるURLは

http://www.any-domain.or.jp/index.html

と同値となっている。

【0140】

実際にアクセスするファイル名となる文字列（（1910）、（1914）参照。）を得たらWWWサーバの制御部（610）は、これを変数情報保持部（618）に格納するほか、ファイルアクセス部（616）に介して目的のファイルへのアクセスを試みる。ファイルアクセス部（616）はファイル保持部（620）のインデックス領域（図8A参照）にアクセスして目的のファイルが格納されている格納場所（pos11、pos12等）を得て、目的のファイルへのアクセスを試みる（図8B参照）。

【0141】

目的のファイルがテキストデータが画像データ等のデータファイルであればこれを読み出す。目的のファイルがCGIプログラムであれば（実行許可等があれば）これを実行する。

【0142】

以上により得られたファイルの中身やプログラムの実行結果を、ヘッダと呼ばれる所定のデータとともにHTTPレスポンスとしてデータ送信部（604）を介してWWWクライアントに送信する。

【0143】

なお、目的のファイルへのアクセスが失敗した場合はエラーとなる。この場合は、エラー処理部（610）にてエラーメッセージを生成し、ヘッダとともにHTTPレスポンスとしてデータ送信部（604）を介してWWWクライアントに送信する。多くの場合ブラウザの受信データ表示欄（504）には、「目的のファイルにはアクセスできません」等の主旨のエラーメッセージが表示されることになる。

【0144】

以上が既存のWWWサーバにおける処理の概要である。

【0145】

以下では、FxURL対応WWWサーバの、この実施の形態において特徴的な事項を説明する。WWWサーバの内部構成や関連データ領域等については図7を、FxURL対応WWWサーバの処理の流れについては図1をそれぞれ参照されたい。この実施の形態のFxURL対応WWWサーバでは、WWWサーバの固定的部位は何ら変更しない。

【0146】

説明のおおよその流れは次のとおりである：

- (1)機能拡張部の準備。
- (2)エラーを誘発するデータの設定。
- (3)エラー発生時の分岐に関する設定。
- (4)機能拡張部等を備えたWWWサーバの起動とHTTPリクエストに対する処理。

【0147】

まず、機能拡張部（702）となる文字列処理手段を準備する。

【0148】

URLパス部を用いるものならどのような処理を行うものでも良いが、例えば、URLパス部の文字列をキーとして外部のデータベースを検索し、当該キーに対応するURLを成す文字列を値として得て、その値を所定のステータスコードとともにWWWクライアントに送信することによりURLの転送処理をさせるような文字列処理手段；

を挙げることができる。以上で機能拡張部（702）の準備は終わる。機能拡張部（702）は、既存のWWWサーバの固定的部位とも非固定的部位とも関係の無く、新規に設けるものである。

【0149】

次にエラーを誘発するデータの設定を行う。

【0150】

意図的にエラーを誘発するためのデータをファイル保持部（620）内のエラー誘発情報保持部（706）に設定する。具体的には、ファイル保持部（620）へのすべてのアクセスがすべてエラーとなるように設定する。この設定は、WWWサーバが実行時に参照するアクセスコントロールファイルなるファイル（このファイルはファイル保持部（620）に格納される。）を用いることにより実現できる。同様の設定は初期化情報保持部（614）を用いて可能であるが、この実施の形態においてはファイル保持部（620）に格納されるタイプのアクセスコントロールファイルを利用する。図7のファイル保持部（620）はエラー誘発情報保持部（706）を含んだ状態を示している。

【0151】

次に、エラー発生時の分岐に関する設定を行う。

【0152】

前記エラー誘発情報保持部（706）に起因してエラーが発生した際の分岐先に関する情報を初期化情報保持部（614）内の分岐先情報保持部（704）に設定する。具体的には、前記機能拡張部（702）が前記分岐先となるように設定する。この設定は、WWWサーバが起動時に参照するコンフィグレーションファイルなるファイル（このファイルはファイル保持部（620）とは別の領域に格納

される。)を用いることにより実現できる。図7の初期化情報保持部(614)は分岐先情報保持部(704)を含んだ状態を示している。

【0153】

以上で準備は完了なので、次に、機能拡張部等を備えたWWWサーバ(つまりFxURL対応WWWサーバ)の起動を行う。

【0154】

なお、この実施の形態においては条件変更部(708)を備えているので、必要に応じて条件変更部(708)を用いてエラー誘発情報保持部(706)や分岐先情報保持部(704)の内容を変えることができる。

【0155】

WWWサーバを起動すると、図1に示すように、初期化処理が行われる。WWWサーバは自らの動作に必要な変数を変数情報保持部(618)に格納する等の処理をしたのち(ステップ(S1))、上記で設定したエラー発生時の分岐先情報を分岐先情報保持部(704)から取り出して変数情報保持部(618)に格納する(ステップ(S2))。

【0156】

以降は、WWWクライアントからのHTTPリクエストを待つ状態に入る(ステップ(S3))。

【0157】

実際にWWWクライアントからのHTTPリクエストがあるところのFxURL対応WWWサーバは、WWWクライアントから得た情報をデータ受信部(602)を介して受信し、URL解釈部(606)を用いてURLパス部等を取得し(ステップ(S4))、URLパス部等を変数情報保持部(618)に格納し、変数情報保持部(618)に格納されたDocumentRoot等の情報(1902)等を用いてアクセス対象とするファイルを決める(ステップ(S5))。

【0158】

このようにして実際にアクセスするファイル((1914)参照)が決まったらファイルアクセス部(616)を介してファイル保持部(620)にある目的のファイルへのアクセスを試みる(ステップ(S6))。しかしこのアクセスは、

意図的に設定された前記エラー誘発情報保持部の内容が原因となって常に（あるいは条件変更部（708）を用いて変更された条件に応じて）エラーとなる（ステップ（S7））。

【0159】

このためWWWサーバの制御部（610）は制御をエラー処理部（612）へと移す。

【0160】

するとエラー処理部（612）は、ステップ（S2）において設定されたエラー発生時の分岐先情報を変数情報保持部（618）の中の分岐先情報保持部（704）から取り出して、指定された分岐先へと分岐させる（ステップ（S9））。

【0161】

これにより制御は機能拡張部（702）となっている前記文字列処理手段に移り、所望の文字列処理が実施される（ステップ（S10））。

【0162】

前記文字列処理手段の処理結果をこのHTTPリクエスト元のWWWクライアントに対しHTTPレスポンスとしてデータ送信部（604）を介して送信すれば（ステップ（S11））このHTTPリクエストに対する処理は終わる。

【0163】

こうしてステップ（S3）に戻り、次のHTTPリクエストを待つことになる。

【0164】

この実施の形態では文字列処理手段の例としてURL転送を挙げたが、これは一例である。例えば、URLパス部に日本語の文を書いてもらい、これを英訳し、その結果をブラウザの受信データ表示欄（504）、URL入力表示欄（502）、または「ステータス行」などと呼ばれているブラウザの下部領域（図示せず）に表示する；という文字列処理手段を実装することも可能である。

【0165】

あるいは、URLパス部の文字列を電子メールの宛先と（短い）本文と見做してその宛先に送信する；という文字列処理手段を実装することも可能である。

【0166】

(実施形態 2)

意図的にエラーを誘発する他の例として次の例を挙げることができる。なお以下では、ほぼ、この実施の形態（実施形態 2）に特有な事項のみ記述する。全体の処理の流れ等は実施形態 1 の記述を参照されたい。

【0167】

本来はa.htmlからz.htmlまでの26のファイルをDocumentRoot下に格納して

http://www.ox.com/a.html、

http://www.ox.com/p.html、

http://www.ox.com/z.html

なるHTTPリクエストに対して正常に

a.html、

p.html、

z.html、

の内容を送信すべきところを、

a.html～c.html

の3のファイルを除く23のファイル（d.html～z.html）を意図的にDocumentRoot下以外のディレクトリ（例えば/httpd/doc2）に格納する；という意図的なエラー誘発手段を採ることもできる。

【0168】

このようにした場合、

d.html～z.html

へのアクセスは一旦エラーとなる。この結果、この処理はエラー処理部を介して機能拡張部へと制御が移る。

【0169】

これにより、例えば、

d.html～z.html

については変数情報保持部（618）に格納されているURLホスト部やブラウザの種別を表す変数等を用いて、アクセス元のWWWクライアントが属するドメイン名やブラウザの種別やバージョンの相違に応じて好適なドキュメントを返す；等

の処理を文字列処理手段として実装することが可能である。一旦はエラーとなったこの処理は、最終的には正常終了としてWWWクライアントへ必要なデータを送信することができる。

【0170】

(実施形態3)

以下では、ほぼ、この実施の形態(実施形態3)に特有な事項のみ記述する。全体の処理の流れ等は実施形態1の記述を参照されたい。

【0171】

意図的にエラーを誘発する他の例として、DocumentRoot下に何らファイルを置かない；とする方法もある。このようにした場合、

- (1)URLパス部が「sales.html」というようなTrURLである文字列の場合も、
 - (2)URLパス部が「営業のご案内.html」というようなFxURLである文字列の場合も、
 - (3)URLパス部が「営業案内.html」というようなFxURLである文字列の場合も、
- すべて

File Not Found

のようなエラーとなる。DocumentRoot下に何らファイルが無いので当然である。既存のWWWサーバの場合であれば、このようなWWWサーバはWWWクライアントに対してファイルを提供するという機能を果たさないが、FxURL対応WWWサーバは違う。制御は一旦エラー処理部(612)に移ったのち、更に、機能拡張部(702)に移る。

【0172】

これにより、WWWクライアントからのファイルアクセスを伴うすべてのHTTPリクエストに対して統一した処理を行うことができ、便利である。上記の例で言えば、URLパス部が

「sales.html」、

「営業のご案内.html」または

「営業案内.html」

のいずれであっても同一のファイルをHTTPレスポンスとして送信するなど、異な

るURLの同一視等の文字列処理が容易に実現できる。

【0173】

また、副次的な効果として、1のディレクトリに収容可能なファイルの数の制限を事実上撤廃することができる。

【0174】

従来であれば、大量のファイル（ディレクトリを含む。）や収容したくてもOSのファイルシステムの仕様その他に起因する制限により1のディレクトリに収容可能なファイルを格納するのは、たとえそれが可能であった場合でも管理面で困難であった。多くのインターネットのプロバイダは会員のためにホームページ用のディレクトリを用意するが、会員数の増大に伴いその管理も大変になっていることであろう。1のコンピュータでは収容しきれないため複数のコンピュータを用いるプロバイダもある。そうすると管理はますます困難でコストの掛かるものとなる。

【0175】

ところがこの実施の形態によればWWWクライアント側（のURLパス部）で指定された階層構造に縛られずにファイルアクセスを伴うすべてのHTTPリクエストに対して外部の大型データベースから必要な情報を得るようなWWWサーバを構築できるので、たとえば、一千万人分のホームページを統一して管理することも比較的容易に実現できる。これは、仮想的に1のディレクトリに一千万人分のホームページを置くことができることを意味している。必要であれば一千万を越える人数分のホームページを置くこともできる。事実上、収容可能なファイルの数の制限を撤廃していると言って良いであろう。

【0176】

念のため従来技術との違いを補足する。上述のような処理を実現する場合、従来でも図13Bのような形式（つまり実質的には図13B2と同じ形式）であるならば実現できるかもしれないが、本発明の場合、図13Aのような形式で実現できる。

【0177】

補足として、この実施の形態の変形として次のような形態を挙げることができる

。すなわち、HTTPリクエストのうち、URLがTrURLであるものについては特別な文字列処理は不要であるとして従来どおりDocumentRoot以下にファイルを格納してもよい。このようにすれば、URLがTrURLであるHTTPリクエストについてはWWWサーバ内の本来の処理部によりすばやく処理する一方、URLが漢字等を含むFxURLであるHTTPリクエストについては機能拡張部にてワイルドカードの解釈を行う等のURLの機能を高めるような処理を行うことが可能となる。

【0178】

なお、条件設定部（708）を用いてファイル保持部（620）中のエラー誘発情報保持部（706）の内容を時間帯によって変更すれば、時間帯によってWWWサーバの振る舞い（エラーを誘発する条件等）をを変更することが可能である。

【0179】

図7に示した各部を備えた処理手段をCPU（1706）を用いた装置のハードウェア構成の一例を図17に示す。なお、図7に示した各部分の全部または一部はCPUを用いずにハードウェアロジックにより構成してもよい。

【0180】

図17において、CPU（1706）には、
ネットワークインターフェース（1702）、
メモリ（1708）、
ディスプレイ（1710）、
キーボード（1712）、
CD-ROMドライブ（1716）、
ハードディスク（1718）
が接続されている。

【0181】

ハードディスク（1718）には、OS（1720）、WWWサーバ装置に必須のプログラムであるhttpdプログラム（1722）、httpdプログラムが動作中に各種設定情報や変数等が格納されるhttpdプログラム用ワークエリア（1724）、httpdプログラムがWWWクライアントからのHTTPリクエストに応じて随時提供する情報等を格納した公開用ファイル（1726）が記憶されている。これらがWWW

サーバの基本的要素である。

【0182】

ハードディスク（1718）にはまた、この発明を用いたWWWサーバにだけ存在する機能拡張プログラム（1728）と機能拡張プログラム用ワークエリア（1730）も記憶されている。これがWWWサーバの拡張的要素である。

【0183】

WWWサーバの基本的要素のうちWWWサーバに特有の要素はhttpdプログラム（1722）とhttpdプログラム用ワークエリア（1724）と公開用ファイル（1726）であるが、これらのうちhttpdプログラム（1722）は通常、修正に際してはソースコードのコンパイルを伴う固定的部位である。ソースコードをコンパイルするには、コンパイラ自体の入手の他にソースコードの入手やハードディスク内への展開、コンパイル作業、（再）インストール作業など、大変工数の掛かる作業となる場合がある。

【0184】

請求項1記載のエラー発生手段、文字列処理手段、分岐手段のうち、すくなくとも1の手段はhttpdプログラム（1722）を何ら変更することなく備えることが可能であれば請求項3を満たす。もしもこの発明を商用WWWサーバに適用可能であり、かつ、その商用WWWサーバのソースコードが提供されないのであれば、それは請求項3を満たすことになる。ソースコードの公開が一般的である「フリーソフト」と呼ばれる類のプログラムであっても、そのプログラムのソースコードを再コンパイルすること無しにこの発明が適用可能であるならば、それは請求項3を満たすことになる。

【0185】

ハードディスク（1718）に記憶されている各要素は、

(1)CD-ROMドライブ（1716）を介してCD-ROM（1714）によりインストールされたか、

(2)ネットワークインターフェース（1702）を介してネットワーク（1704）上にあるWWWサーバまたはFTPサーバ（いずれも図示せず）からインストールされたものである。

【0186】

WWWサーバのデータ受信部（1602）はネットワークインターフェース（1702）とOS（1720）を介してWWWクライアントから送られて来るHTTPリクエストを受信する。

【0187】

WWWサーバのデータ送信部（1604）はOS（1720）とネットワークインターフェース（1702）を介してHTTPリクエストに対する応答となるデータ（HTTPレスポンス）をWWWクライアントに対して送信する。

【0188】

WWWサーバのURL解釈部（606）は、URLを分解する際にCPUにて演算を実行する他、URL分解処理の結果得られるURLパス部の文字列のデータをハードディスク（1718）内のhttpdプログラム用ワークエリア（1724）における変数としてまたはメモリ（1708）内におけるデータとして記憶または一時記憶させる。

【0189】

WWWサーバの初期化処理部（608）が参照する初期化情報保持部（614）はhttpdプログラム（1722）の初期化ファイル（別称：コンフィグレーションファイル）として実現される。このファイルはハードディスク（1718）内のhttpdプログラム用ワークエリア（1724）に記憶される。

【0190】

初期化ファイルの修正に際しては、このWWWサーバについての権限ある操作者（図示せず）がOS（1720）の一部として備えられているエディタ（テキストファイル編集プログラム）等を用いて初期化ファイルの内容をディスプレイ（1710）に表示させつつキーボード（1712）を用いてキー入力を行って当該ファイルの内容の書き換えを行うことができる。

【0191】

なお、ディスプレイ（1710）やキーボード（1712）を用いずに、ネットワークインターフェース（1702）を介してネットワーク（1704）に接続されている他のコンピュータからファイル転送等の手段によっても初期化ファイ

ルの内容の書き換えは可能である。

【0192】

WWWクライアントからのHTTPリクエストに応じて送信するHTMLファイル（テキストファイル）やGIFファイル（画像データ）等のファイルはDocumentRoot下に置かれ、CGIプログラムは通常ScriptAlias下またはDocumentRoot下に置かれる。これらはいずれも、図7においてはファイル保持部（620）に、図17においては公開用ファイル（1726）に含まれる。CGIプログラムの一部または全部は（1726）の外に置かれる場合もある。

【0193】

WWWサーバが通常の動作に必要な要素の説明は以上のとおりである。

【0194】

加えてこの発明を実施するWWWサーバのある実施の形態では、意図的にエラーを誘発するためのエラー誘発情報保持部（706）がファイル保持部（620）の一部としてファイルの形で挿入されている。このファイルはアクセスコントロールファイル等と呼ばれている。図17でいえば、公開用ファイル（1726）の一部として記憶されていることになる。

【0195】

この発明を実施するWWWサーバの別の実施の形態では、エラー誘発情報保持部（706）は初期化情報保持部（614）に挿入されている。この場合、図17でいえば、httpdプログラム用ワークエリア（1724）に記憶されていることになる。

【0196】

加えてこの発明を実施するWWWサーバでは、エラー発生時の分岐先を決定するための情報を分岐先情報保持部（704）を備えるが、この情報の素となるデータ片は初期化情報保持部（614）に挿入されていることとし、初期化処理部（608）の処理により変数情報保持部（618）に記憶されることとする。この場合分岐先情報の素となるデータ片は初期化ファイル中に存在するので、図17でいえばhttpdプログラム用ワークエリア（1724）に記憶されていることになる。

【0197】

もっとも、請求項3を満たさなくてもよいのであれば、httpdプログラム（1722）のソースコードを修正してhttpdプログラム（1722）の内部に分岐先情報保持部（1614）に保持すべきデータを置くこともできるし、後述する機能拡張プログラム（1728）で実現可能な機能をhttpdプログラム（1722）内に取り込むこともできる。

【0198】

加えてこの発明を実施するWWWサーバでは、WWWサーバの機能を拡張するための機能拡張部（612）を備える。機能拡張部（612）は、機能拡張プログラム（1728）と機能拡張プログラム用ワークエリア（1730）としてハードディスク（1718）内に記憶されている。

【0199】

加えてこの発明を実施するWWWサーバのうち請求項2を満たすものは、分岐先情報保持部（704）の内容を変更するための手段として条件変更部（708）を備える。条件変更部（708）は前記アクセスコントロールファイルまたは同等機能を提供するファイル等における分岐先情報保持部（704）またはこれに相当する箇所を修正するためのものである。

【0200】

上記実施形態ではCD-ROMドライブ（1716）、ディスプレイ（1710）およびキーボード（1712）を用いたが、ハードディスク（1718）内へのファイルのインストール、ハードディスク（1718）内のファイルの書き換え等の作業、httpdプログラム（1722）やWWWサーバ全体の監視・管理等のすべての作業をネットワークインターフェース（1702）を介してネットワーク（1704）に接続可能な他のコンピュータ（図示せず）から遠隔処理（リモート処理）するのであれば、CD-ROMドライブ（1716）、ディスプレイ（1710）およびキーボード（1712）は備える必要はない。

【0201】

【発明の効果】

これまでに説明から明らかなように、WWWサーバにおけるURLの解釈を柔軟に行う

ことによりURLの表現力や利用価値を大幅に高めることができる。同時に、インターネットのサービスを利用する際の障壁を下げる効果もある。これらの相乗効果により現在のインターネット利用者にも将来のインターネット利用者にもメリットをもたらすサービスが提供可能である。また、WWWサーバ以外にも応用可能であるため多分野に渡り効果を得ることができる。

【図面の簡単な説明】

【図 1】 この発明の一実施形態による機能拡張手段等を備えたWWWサーバの処理の流れを示す図である。

【図 2】 WWWサービスの一般的な利用形態を示す図である。

【図 3】 WWWサービスにおけるクライアントとサーバ間でのやり取りを単純化した図である。

【図 4】 URLの構成要素を説明した図である。

【図 5】 ブラウザの画面インターフェースを単純化した図である。

【図 6】 既存のWWWサーバの構成要素と関連データを示すブロック図である。

【図 7】 この発明の一実施形態による機能拡張手段等を備えたWWWサーバの構成要素を示すブロック図である。

【図 8】 ファイルシステムにおけるインデックス領域とファイル領域との関連を示した図である。

【図 9】 ○×商会のホームページのURL（URLは通常のTrURLである。）と内容を示す図である。

【図 10】 ○×商会のホームページのURL（URLは漢字等を含んだKURLである。）と内容を示す図である。

【図 11】 URLの表記における各ASCII文字について利用の適不適を示す図である。

【図 12】 末尾に記号「?」を含むURLの例を示す図である。

【図 13】 URL中の記号「?」の有無とURLの表現能力との関係を示すための図である。

【図 14】 この発明により可能になるURL表現の例を利用可能な文字種の豊

富さに力点を置いて示した図である。

【図 15】この発明により可能になる URL 表現の例を検索機能に力点を置いて示した図である。

【図 16】この発明により可能になる URL 表現の例を日常の文章表現が可能な点に力点を置いて示した図である。

【図 17】この発明の一実施形態による機能拡張手段等を備えた WWW サーバ処理装置の構成要素を示す図である。

【図 18】与えられた URL を URL パス部など 4 つの部分に分解する URL 分解手段の機能を示す図である。

【図 19】WWW サーバの初期設定ファイルの内容と WWW クライアントから得た URL パス部とがどのように作用して実際のアクセス対象ファイルを決めるのかを示す図である。

【符号の説明】

- (S 1) 初期化処理ステップ (一般的な設定)
- (S 2) 初期化処理ステップ (この発明に特有な事項)
- (S 3) HTTP リクエスト受信判断ステップ
- (S 4) URL パス部獲得ステップ
- (S 5) アクセス対象ファイル決定ステップ
- (S 6) アクセス対象ファイルへのアクセスステップ
- (S 7) エラー発生ステップ
- (S 8) エラー処理ステップ
- (S 9) 文字列処理ステップへの分岐ステップ
- (S 10) 文字列処理ステップ
- (S 11) HTTP レスポンス送信ステップ
- (202) WWW クライアント
- (204) 小規模ネットワーク (LAN、イントラネット等)
- (206) WWW 中継処理装置
- (208) 広域ネットワーク (インターネット等)
- (210) WWW サーバ

- (502) ブラウザのURL入力表示欄
 - (504) ブラウザの受信データ表示欄
 - (602) WWWサーバのデータ受信部
 - (604) WWWサーバのデータ送信部
 - (606) WWWサーバのURL解釈部
 - (608) WWWサーバの初期化処理部
 - (610) WWWサーバの制御部
 - (612) WWWサーバのエラー処理部
 - (614) WWWサーバの初期化情報保持部
 - (616) WWWサーバのファイルアクセス部
 - (618) WWWサーバの変数情報保持部
 - (620) WWWサーバのファイル保持部
 - (702) WWWサーバ用の機能拡張部 (この発明による新規部分)
 - (704) WWWサーバ用の分岐先情報保持部 (この発明による新規部分)
 - (706) WWWサーバ用のエラー誘発情報保持部 (この発明による新規部分)
 -)
 - (708) WWWサーバ用の条件変更部 (この発明による新規部分)
 - (901) ○×商会のトップページ (URLは従来のTrURL) のURL入力表示欄
 - (902) ○×商会のトップページ (URLは従来のTrURL)
 - (904) ○×商会のトップページ (URLは従来のTrURL) における営業部のページへのリンク部
 - (906) ○×商会のトップページ (URLは従来のTrURL) における技術部のページへのリンク部
 - (907) ○×商会の営業部のページ (URLは従来のTrURL) のURL入力表示欄
-
- (908) ○×商会の営業部のページ (URLは従来のTrURL)
 - (909) ○×商会の技術部のページ (URLは従来のTrURL) のURL入力表示欄
 - (910) ○×商会の技術部のページ (URLは従来のTrURL)

(1001) ○×商会のトップページ (URLは漢字等を含むKURL) のURL入力
表示欄

(1002) ○×商会のトップページ (URLは漢字等を含むKURL)

(1004) ○×商会のトップページ (URLは漢字等を含むKURL) における
営業部のページへのリンク部

(1006) ○×商会のトップページ (URLは漢字等を含むKURL) における
技術部のページへのリンク部

(1007) ○×商会の営業部のページ (URLは漢字等を含むKURL) のURL入
力表示欄

(1008) ○×商会の営業部のページ (URLは漢字等を含むKURL)

(1009) ○×商会の技術部のページ (URLは漢字等を含むKURL) のURL入
力表示欄

(1010) ○×商会の技術部のページ (URLは漢字等を含むKURL)

(1102) 非図形文字一覧

(1104) Unsafeな文字一覧

(1106) Reservedな文字一覧

(1108) 制約無く使用可能な文字一覧

(1702) WWWサーバ処理装置のネットワークインターフェース

(1704) ネットワーク

(1706) WWWサーバ処理装置のCPU (中央演算処理装置)

(1708) WWWサーバ処理装置のメモリ

(1710) WWWサーバ処理装置のディスプレイ

(1712) WWWサーバ処理装置のキーボード

(1714) WWWサーバプログラムを記録したCD-ROM

(1716) WWWサーバ処理装置のCD-ROMドライブ

(1718) WWWサーバ処理装置のハードディスク

(1720) ハードディスク内のOS

(1722) ハードディスク内のhttpdプログラム

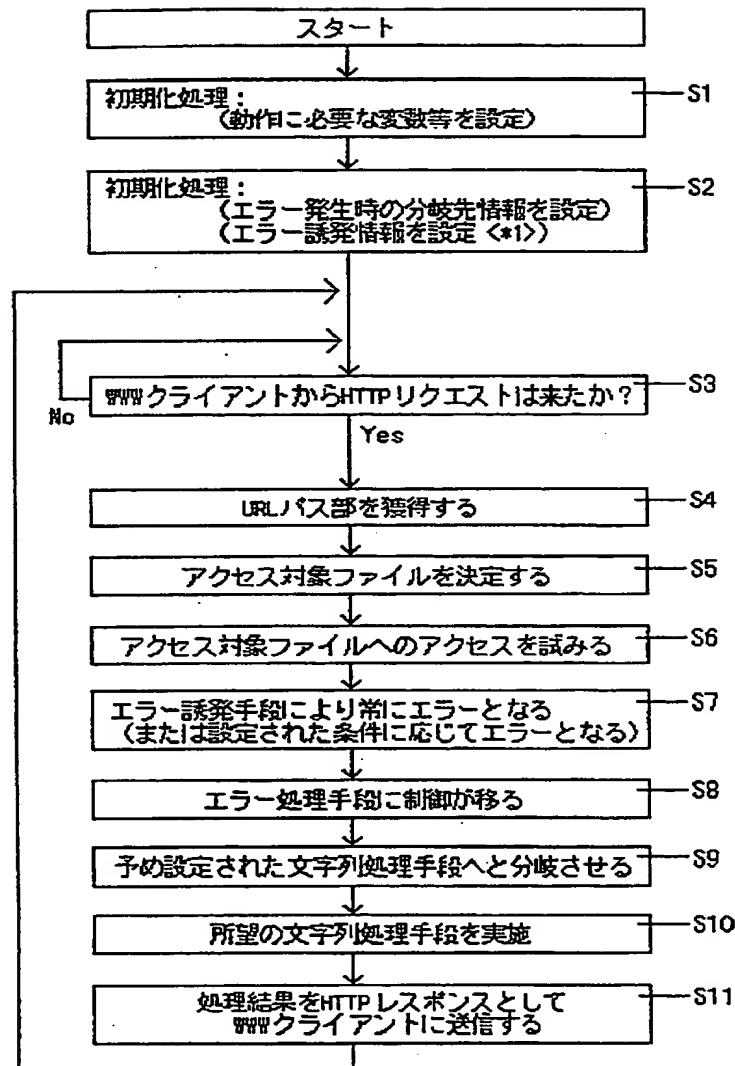
(1724) ハードディスク内のhttpdプログラム用ワークエリア

- (1726) ハードディスク内の公開用ファイル
- (1728) ハードディスク内の機能拡張プログラム
- (1730) ハードディスク内の機能拡張プログラム用ワークエリア
- (1802) 分解前のURL
- (1804) URL分解手段
- (1806) URL分解手段により分解されたURLプロトコル部
- (1808) URL分解手段により分解されたURLホスト部
- (1810) URL分解手段により分解されたURLポート部
- (1812) URL分解手段により分解されたURLパス部
- (1902) 変数保持部のデータの一部（アクセス対象ファイル決定関連）
- (1906) WWWクライアントからのURLパス部の文字列
- (1908) アクセス対象決定手段
- (1910) アクセス対象ファイルを指す文字列（フルパス）
- (1912) URLパス部（（1906）の例）
- (1914) 実際にアクセスするファイル（（1912）の例に対応）

【書類名】

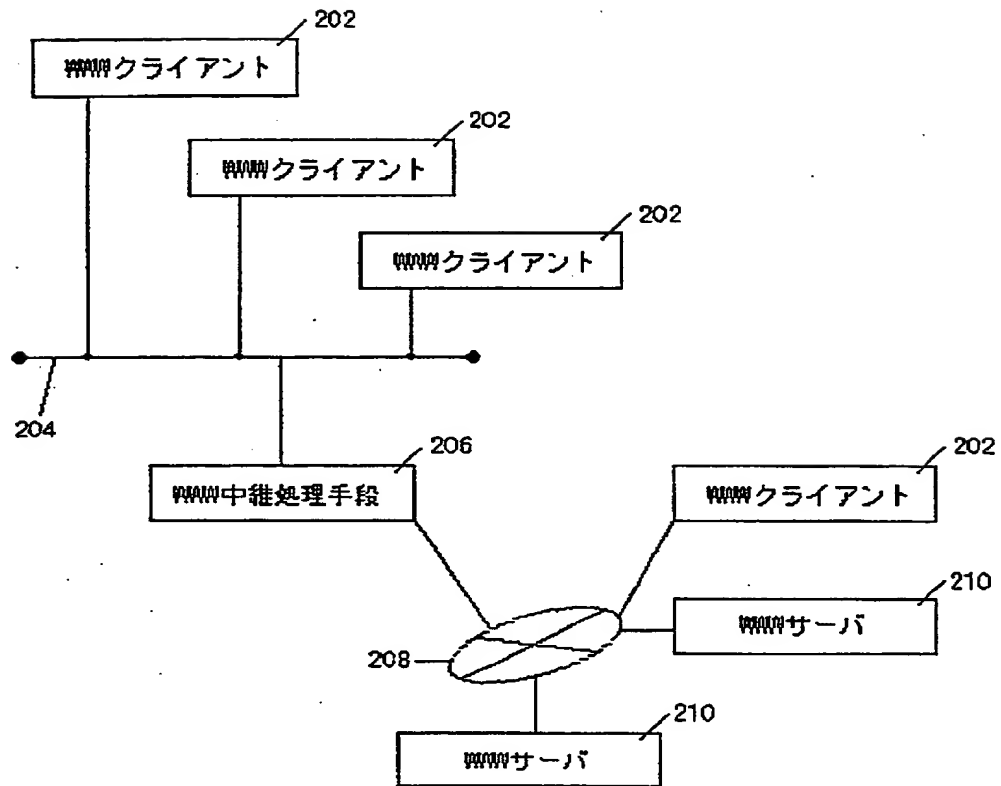
図面

【図 1】

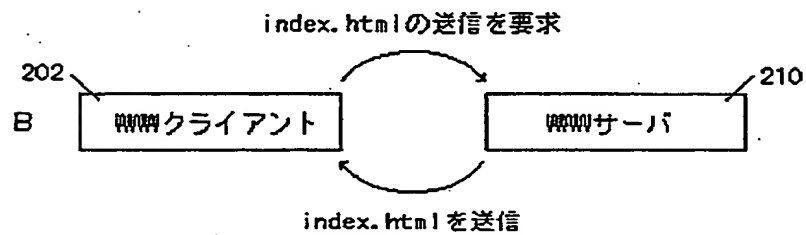
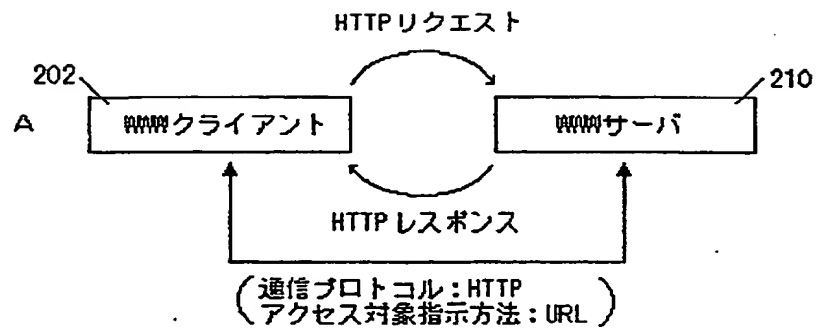


*1 ファイル保持部 (620) にエラー誘発情報を設定する場合は
初期化処理においてはこの設定を行わなくてもよい

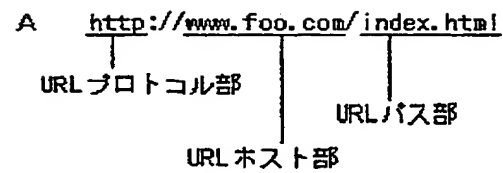
【図 2】



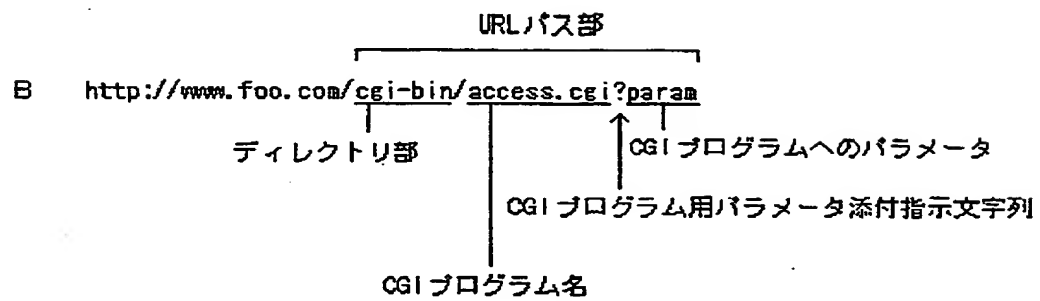
【図 3】



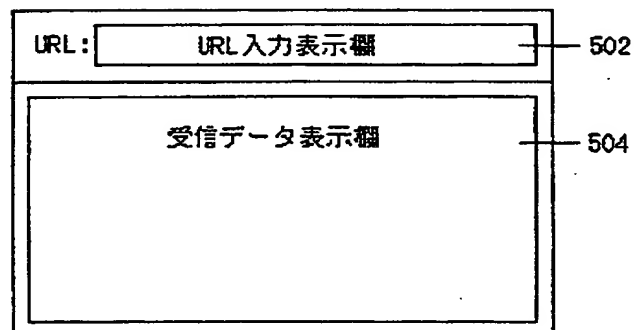
【図4】



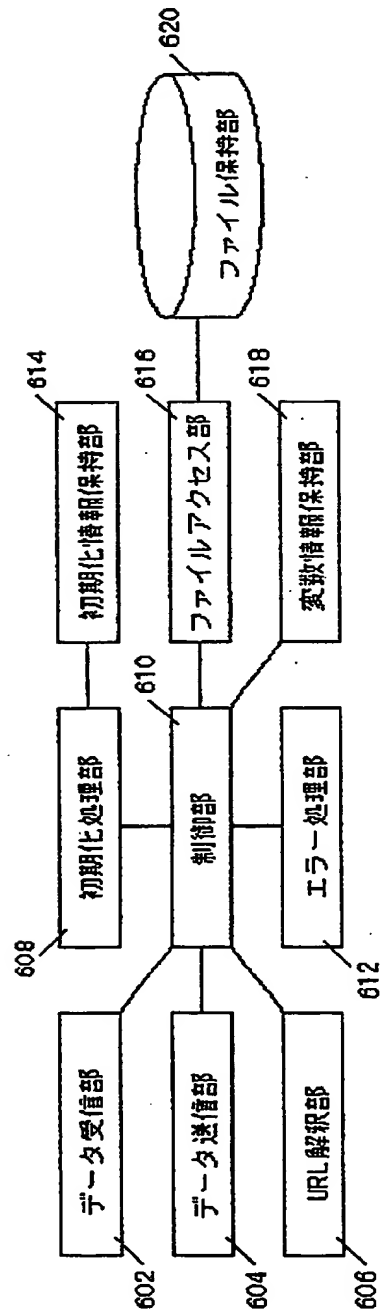
A2 `http://www.foo.com:80/index.html`
 URLポート部



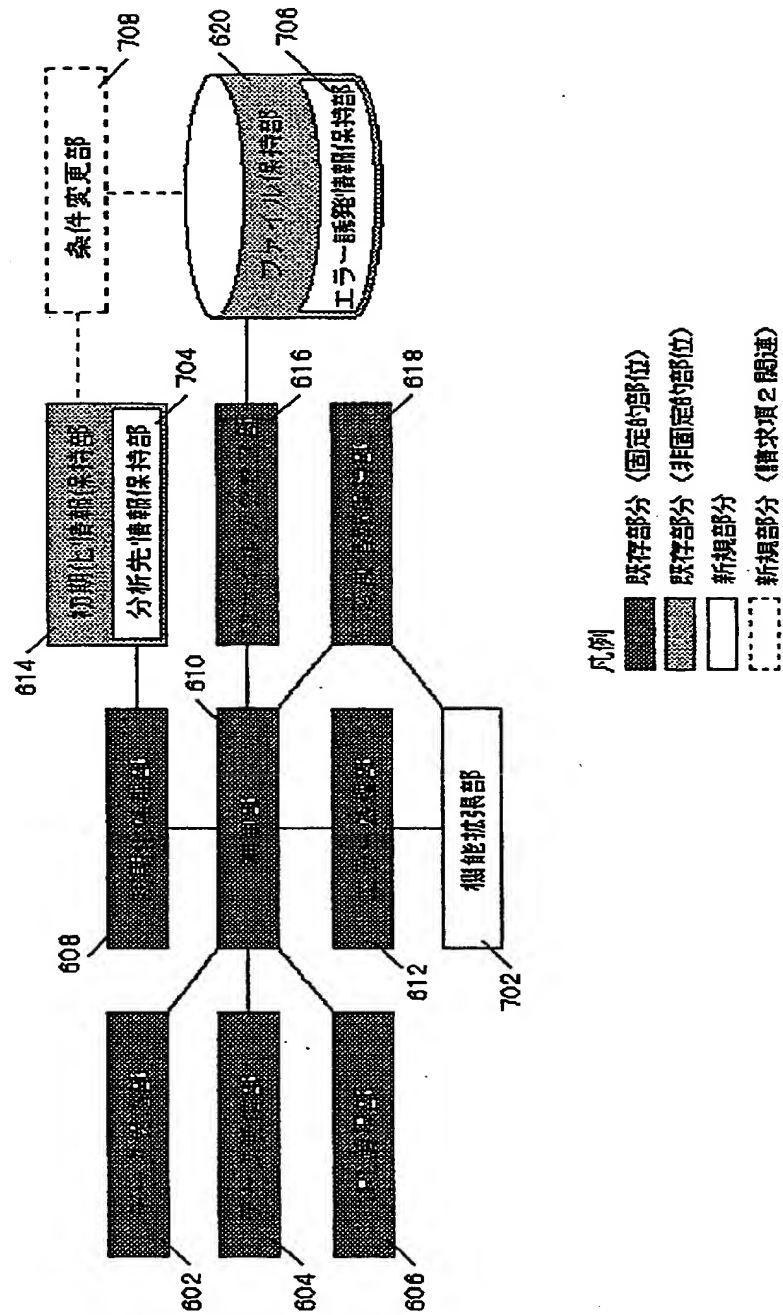
【図5】



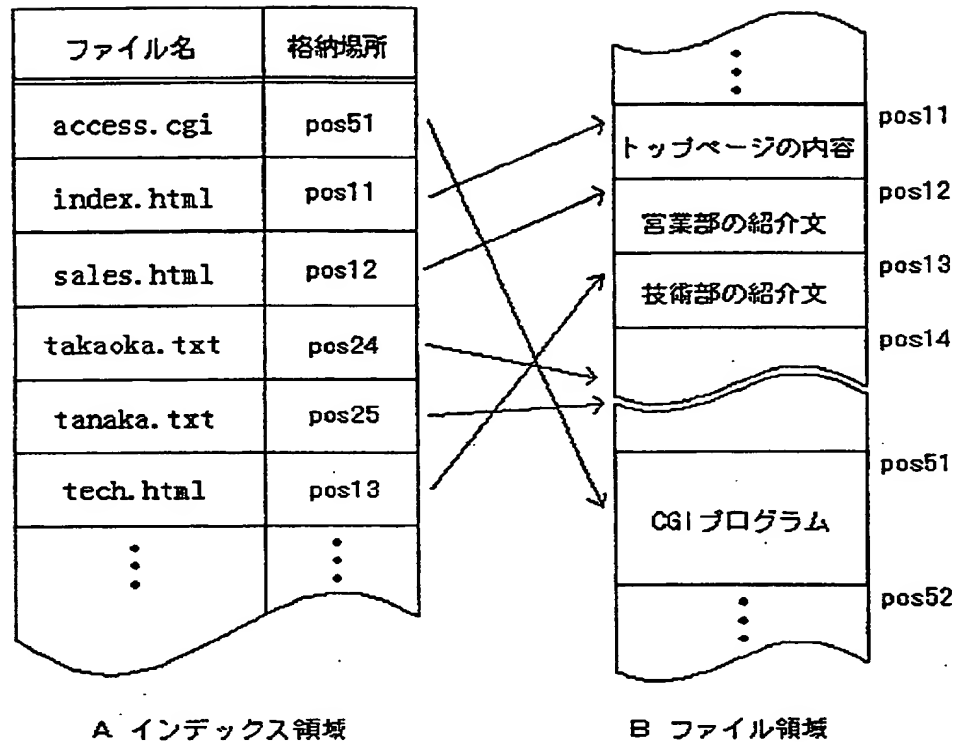
【図6】



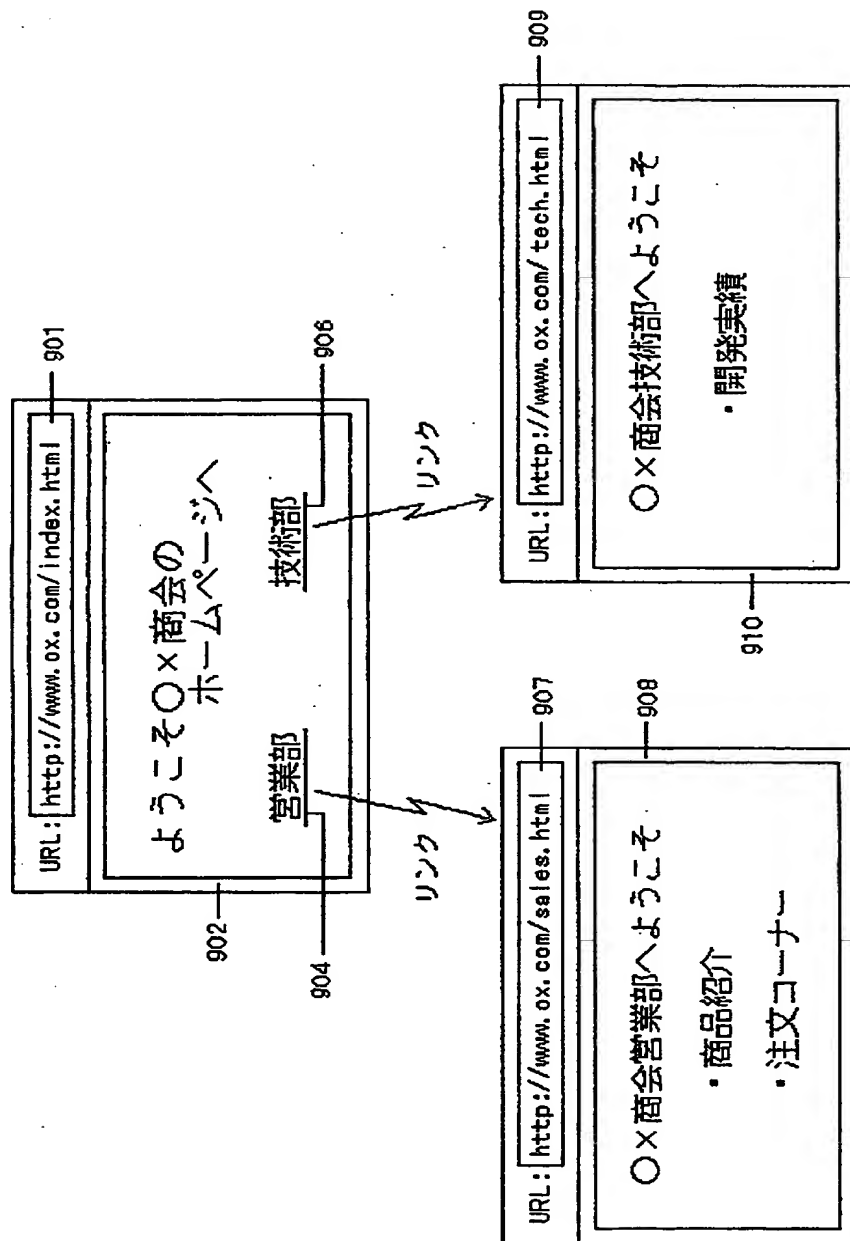
【図 7】



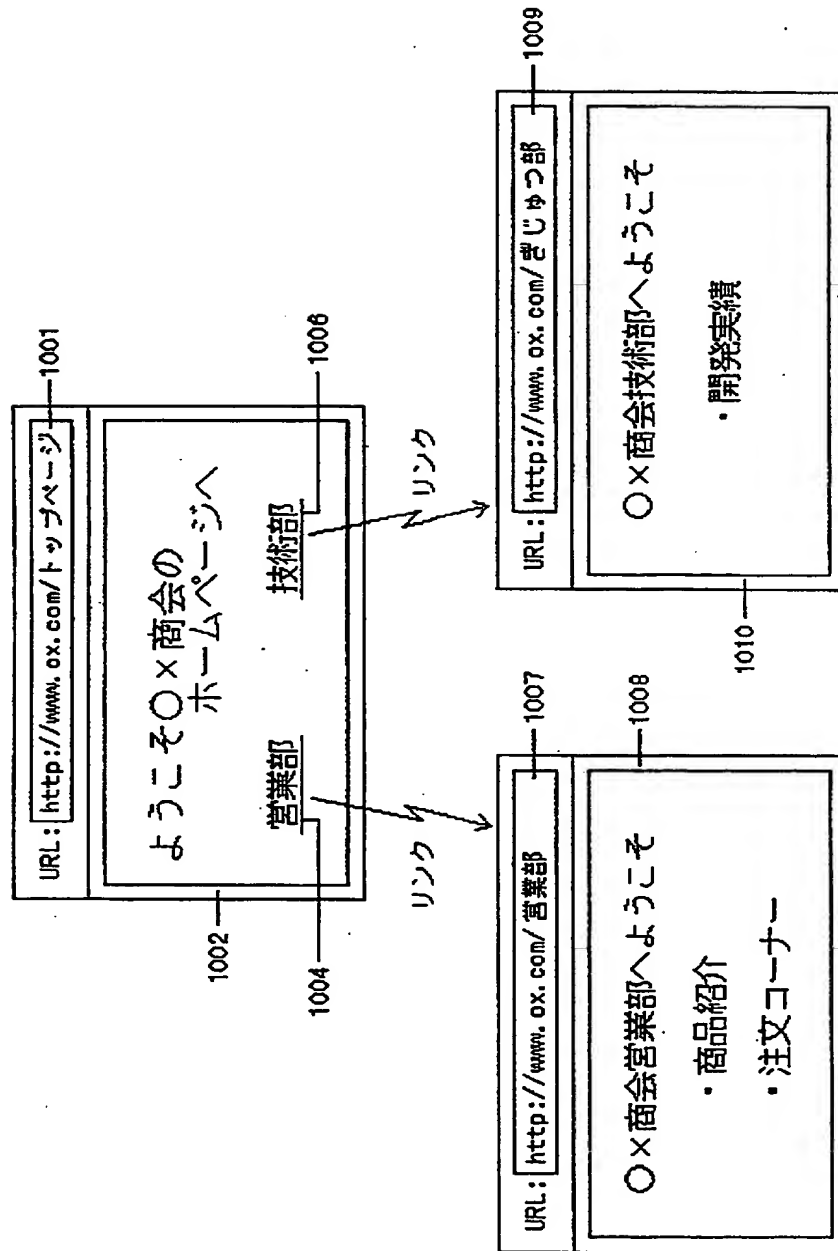
【図 8】



【図 9】



【図 10】



【図11】

非図形文字	1102
0x00~0x1F 0x7F	(33文字)
Unsafeな文字	1104
△ < > “ # % [] ￥ ^ ~ [] 、	(15文字)
Reservedな文字	1106
; / ? : @ = &	(7文字)
制約なく使用可能な文字	1108
A~Z a~z 0~9 \$ - _ . + ! * ' () ,	(73文字)
(ASCII文字コード集合全体: 128文字)	

※△ = Space = 0x20

※非図形文字は16進数表示とした。

※出典: RFC 1738 “Uniform Resource Locators (URL)”

【図12】

A URL: `http://www.ox.com/sales/How-much?`

B URL: `http://www.ox.com/sales/How-much`

C URL: `http://www.ox.com/sales/How△much?`

凡例:△-US-ASCIIの空白文字。
C言語の16進数表記で0x20。
いわゆる「半角スペース」。
※他の図でも同様。

【図13】

A `http://my.name.is/` 林毅

B `http://my.name.is/?` 林毅

B2 `http://my.name.is/index.cgi?` 林毅

C `http://your.name.is/` 林毅?

【図 14】

- A <http://HelloWorld.net/Japanese/> こんにちは
- B <http://HelloWorld.net/English/Hello>
- C <http://HelloWorld.net/French/Bonjour>
- D <http://HelloWorld.net/Chinese/> 你好
- E <http://HelloWorld.net/C/main0> {printf("Hello");}
- F <http://any-data.net/mixed/> これは△pencil△です。
- G [http://any-data.net/math/y=f\(x\)=x^2-3x+2=0](http://any-data.net/math/y=f(x)=x^2-3x+2=0)
- H <http://any-data.net/iso-8859-1/compte> à la fois
- I <http://any-data.net/non-standard-symbol/> 高
- J <http://any-data.net/bitmap/> 
- K <http://any-data.net/encrypted/af0kzf26:9s++B>
- L <http://any-data.net/mailto:info@tel.to/> 資料下さいV/fo@bar.com
- L2 <http://any-data.net/mailto:田中@九州/> 元気ですか?//林@横浜

【図 15】

- A <http://Lookup.to/files/a?z.html>
- B http://Lookup.to/files/a*z.html
- C [http://Lookup.to/files/\[0-9\]\[0-9\]*](http://Lookup.to/files/[0-9][0-9]*)
- D http://Lookup.to/DB:names.jp/田中*
- E http://Lookup.to/DB:names.jp/*子
- E2 <http://Lookup.to/DB:names.jp/> 「子」で終わる人名
- F <http://weather.info/DB:アメダス/> 明日午前の横浜南部の天気は?
- F2 <http://weather.info/> 明日午前の横浜南部の天気は?

【図16】

A <http://AnyString.net/answer=3/4=0.75>

B <http://www.ox.com/社員の中の営業部の佐藤さんのプロフィール>

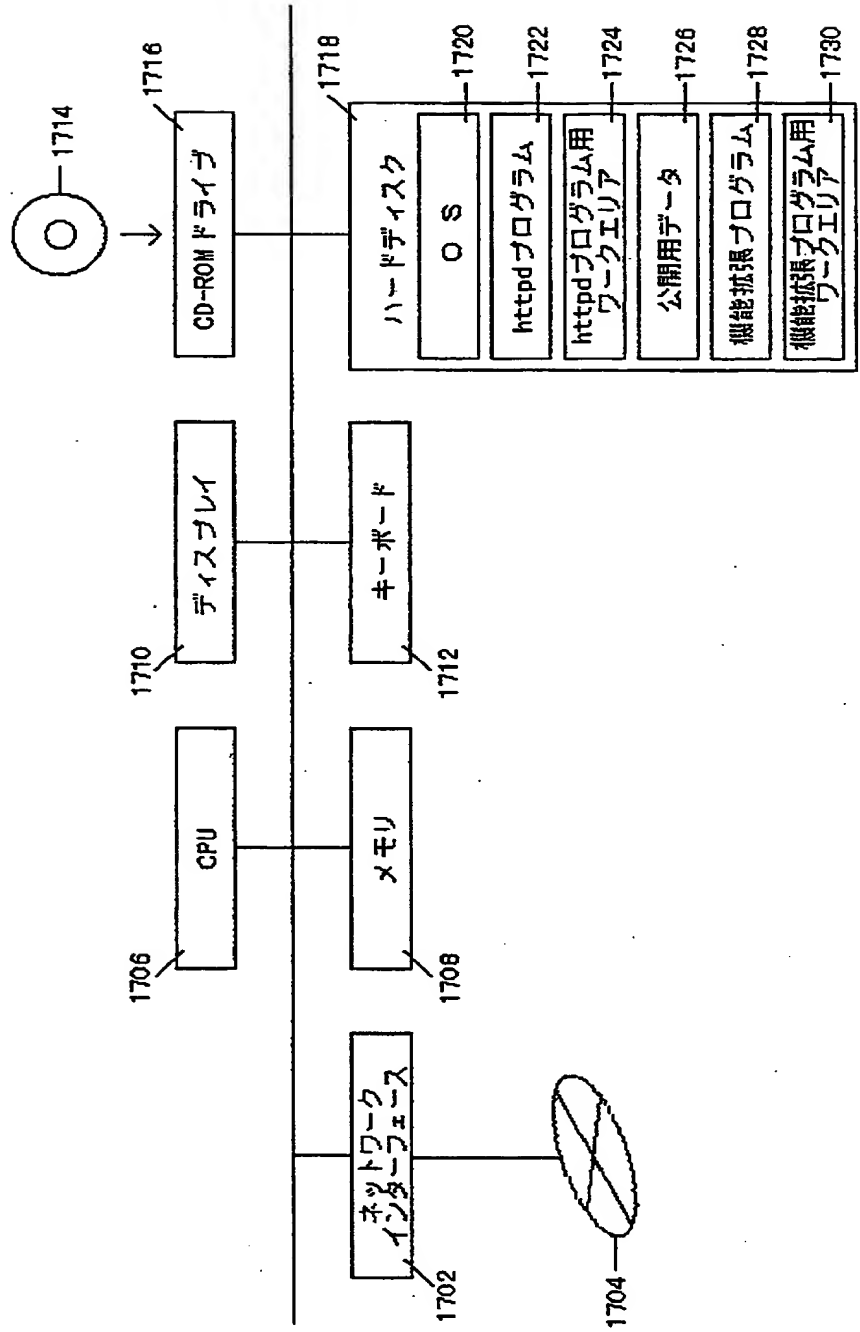
B2 <http://www.ox.com/社員/営業部/佐藤さん/プロフィール>

B3 <http://www.ox.com/associates/sales/Sato/profile>

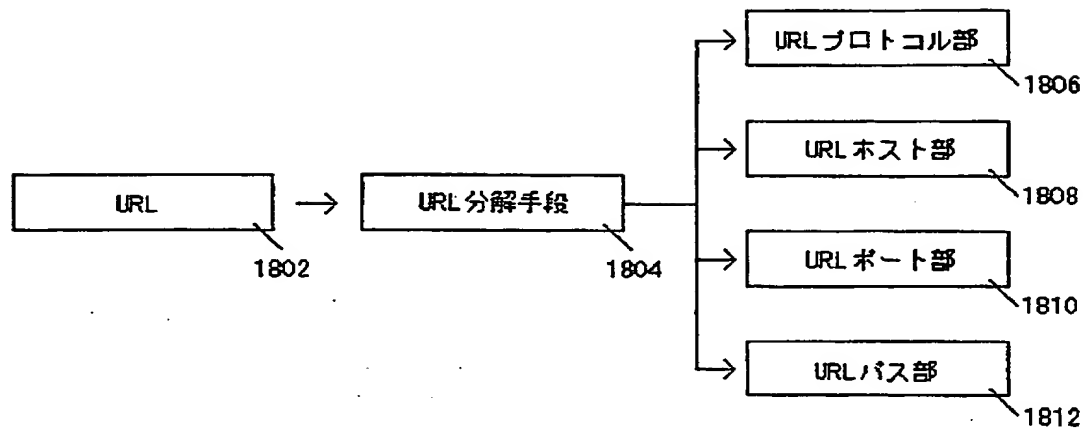
B4 <http://www.ox.com/profile△of△Sato△of△sales div.△of△associates>

※下線部は着目箇所。

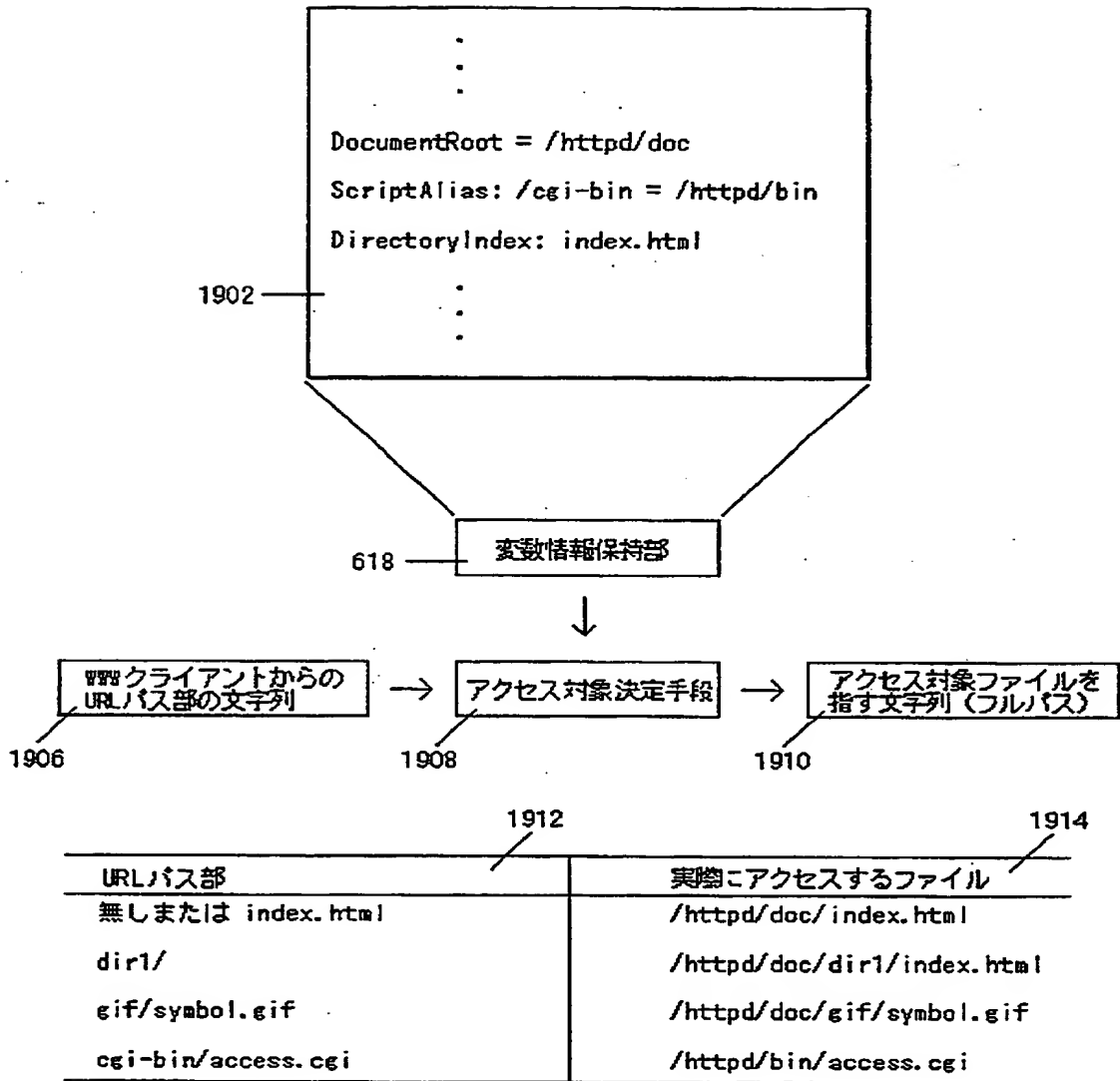
【図 17】



【図 18】



【図 19】



【書類名】 要約書

【要約】

【課題】 HTTPやURLの規約に準拠したこれまでのURLではURLパス部に漢字等の全角文字を含むことができなかった。ASCII文字についても利用上の制約があった。そのため、漢字、ひらがら、カタカナ、記号等を含むURLを用いたくてもできなかったもしくは困難であった。多大な工数を掛けずにこれを実現することを課題とする。当該技術のWWWサーバ以外への広範囲な応用についても併せて課題とする。

【効果】 URLに漢字等を含むことを可能にすることにより、URLの告知手段としての価値を大幅に向上させることができる。インターネット利用者の増大等も見込むことができ、産業の発展に寄与する。

【解決手段】 従来のWWWサーバに、意図的にエラーを発生させるエラー誘発手段や文字列処理手段等を備えた機能拡張装置を備えることにより、任意の図形文字をURLパス部に含むことが可能になる。

【選択図】 図1

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】 申請人
【識別番号】 596151788
【住所又は居所】 神奈川県横浜市磯子区中原二丁目13番19号フラ
ット磯子201号
【氏名又は名称】 有限会社バリアフリー

出 願 人 履 歴 情 報

識別番号 [596151788]

1. 変更年月日 1996年10月 1日

[変更理由] 新規登録

住 所 神奈川県横浜市磯子区中原二丁目13番19号フラット磯子2
01号

氏 名 有限会社バリアフリー

THIS PAGE BLANK (USPTO)